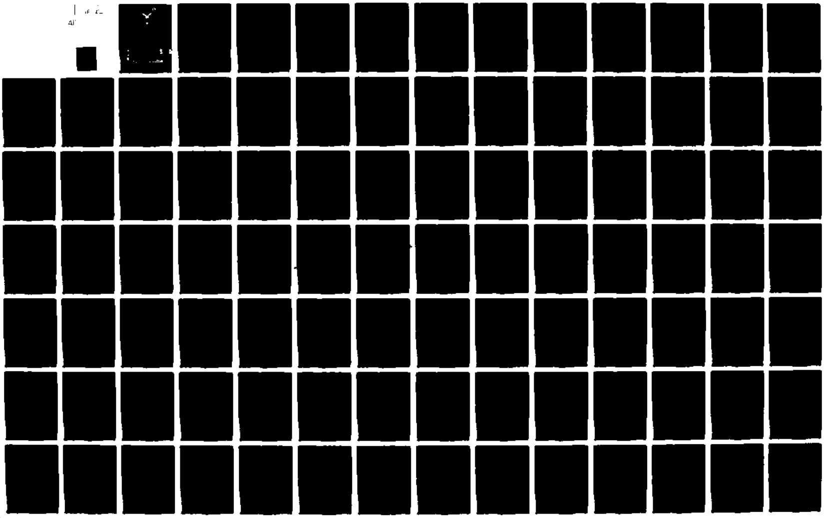
AD-A103 378 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
AN EVALUATION OF DETAIL IN DYNAMIC VISUAL DISPLAYS.(U)
DEC 80 M A SMART
UNCLASSIFIED AFIT/GCS/EE/80-14 NL



LEVEL II

This document has been approved for public release and sale; its distribution is unlimited.

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/EE/80-14

AN EVALUATION OF DETAIL
IN
DYNAMIC VISUAL DISPLAYS.

THESIS

AFIT/GCS/EE/80-14 Mary A. Smart
2Lt USAF

Approved for public; distribution unlimited.

AFIT/GCS/EE/80-14

AN EVALUATION OF DETAIL
IN
DYNAMIC VISUAL DISPLAYS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Mary A. Smart, B.A.

2Lt USAF

Graduate Electrical Engineering

December 1980

Approved for public release; distribution unlimited.

PREFACE

The Information, Presentation, and Controls Group (AAAT-1) of the Air Force Avionics Laboratory is interested in determining the human engineering problems that may be encountered by those using a display with a constantly changing background such as the Airborne Electronic Terrain Mapping System (AETMS). This research was conducted to enhance the AETMS be adding a symbol producing software overlay and to develop the basis for a symbol set for the AETMS.

I would like to extend my thanks to Dr. Matthew Kabrisky, my thesis advisor, Lt. David Rall for his software consultation, and Bruce for his understanding and support.

CONTENTS

List of Figures

List of Tables

ABSTRACT

This report has its basis in the Airborne Electronic Terrain Mapping System (AETMS), an aircraft mapping system based on an idea proposed by L.A. Tamburino and developed by the Air Force Avionics Laboratory.

A constantly changing background provides the basis for an interesting investigation on the hypothesis that a dynamic background will distract the operator's attention away from important details on the display.

The software developed provides an overlay of symbols onto the terrain map generated by the AETMS and is contained in this report. A symbol and color set is suggested for the AETMS.

An experiment to determine a master symbol and color set is suggested. The experiment suggests using both simple vector symbols and Fourier transformed symbols to help define the master set.

An Evaluation of Detail in Dynamic Visual Displays

I. INTRODUCTION

This thesis has two main objectives. One objective is to provide a software package that will overlay symbols onto a terrain map generated by the Airborne Electronic Terrain Mapping System (AETMS). This system was developed by the Information, Presentation, and Controls Group (AAAT-1) of the Air Force Avionics Laboratory, based on an idea proposed by L.A. Tamburino (Ref 16). The software package will provide a video image processor the ability to display symbols as well as terrain. The second objective is to experiment with a small set of symbols to find shapes and colors that are easy to identify.

The hardware basis for this thesis is the AETMS. This system is designed to aid pilots in low alitude flying. "The AETMS is more than a horizonital or vertical situation map system. It is an integrated information system that will supplement current aircraft systems, giving the pilot the capability to negotiate low level, high speed profiles" (Ref 4:10). Designers have constructed software and hardware that converts a digitized representation of terrain into visual display. The software developed for this thesis provides an overlay of a symbol onto the terrain at the appropriate latitude and longitude.

There have been many studies of symbols and displays in the past. These studies were mainly concerned with displays that had non-dynamic or static backgrounds, i.e., backgrounds that do not change. The AETMS display changes as does the vehicle's position. A constantly changing display presents some interesting questions:

1. Will the dynamic background distract the operator's attention away from important details, such as an approaching target?

2. If distraction may occur, then what is the best attention getting shape and/or color for a symbol?

3. What background color should be combined with a symbol's shape and color? These questions are the theoritical basis for the thesis.

Due to system limitations, it was impossible to integrate the thesis software with the AETMS. The reasons for this were twofold. First, the system allows only 28K of memory to be allocated to a task. The thesis software alone took over 28K and finally had to be tasked in three separate packages so that it could be tested by itself. Presently, the thesis software can generate symbols on a plain background. The thesis can be re-written to be overlayed onto itself. This method will allow the thesis software to be tasked with the AETMS. Secondly, the AETMS is a poorly documented system. The system exits upon encountering an error. Explanations as to why an exit occurred are not generated by the system. This makes trouble shooting rather difficult. This system is now being updated to include error statements.

These limitations made it impossible to complete the experiments associated with determining optimum symbol shape and color. Therefore, an explanation of the experiment is included, along with an analysis of symbols and color, a discussion of software, and a literature review.

II. SOFTWARE

The Airborne Electronic Terrain Mapping System (AETMS) can be simply described in the following manner: "The Digital recordings of the Defense Mapping Agency (DMA) world wide data base are preprocessed as depicted to obtain polynomial fits for the terrain elevations (Figure 1). These polynomials are stored as sequences of coefficients (i.e., compressed form). The on-board mass memory of the AETMS will be loaded from a world wide base. The aircraft naviagation system would provide present position inputs to the on-board computer which uses them to access memory. The retrieve coefficients would be used to generate a terrain relief display for the pilot/navigator" (Ref 10:8).

The AETMS is currently being tested on a PDP 11/45. The information generated by the AETMS is displayed on a Ramtek display system via a device driver written by SRL, Inc. The original AETMS was written by the Systran Corporation of Dayton, Ohio. The AAAT-1 group is presently re-writting the AETMS software so that is easier to use and more responsive to the user. Figure 2 shows the system configuration.

The AETMS consists of two software packages, PRSPCT and CNTOUR. PRSPCT provides the user with a direct heads up view of the terrain as seen from the cockpit. CNTOUR provides a display much like that seen on a topographical map. Both PRSPCT and CNTOUR are interactive systems,

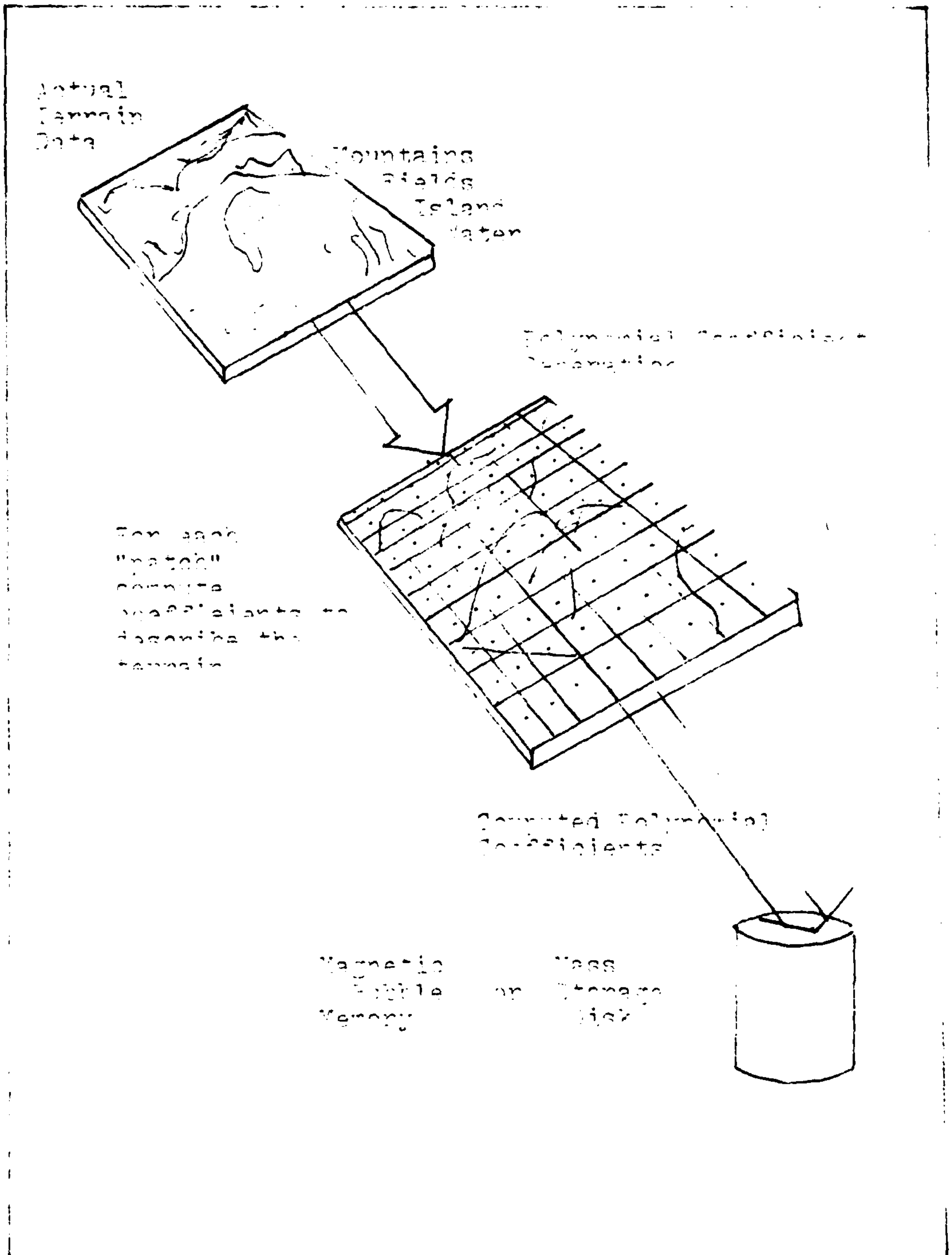


Fig. 1 Data Base Modeling Technique (Ref 11:15)

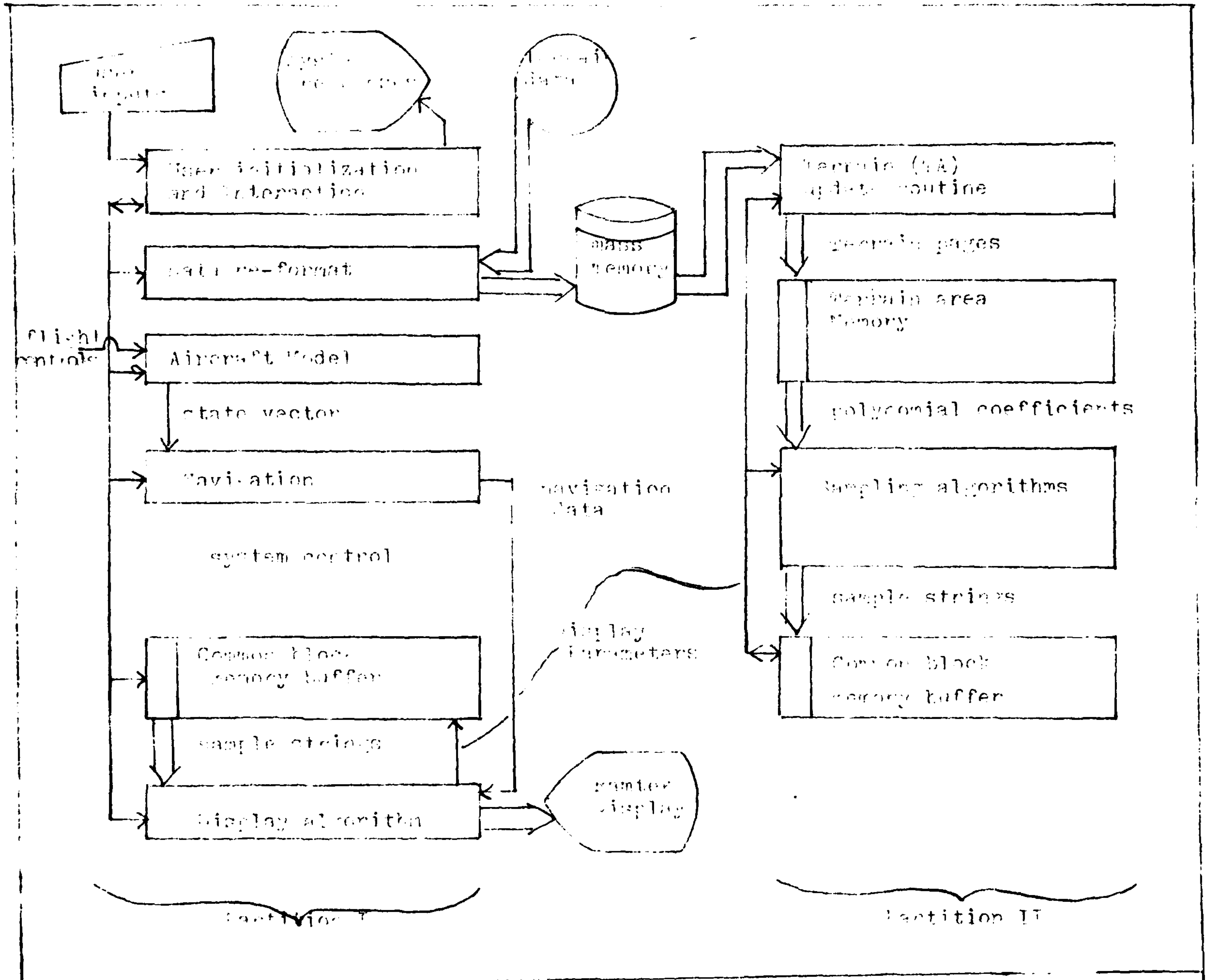


Fig. 2. configuration (of 12)

accepting input from a joystick The interested reader is referred to Systran documentation for a more detailed discussion of PRSPCT and CNTOUR (Ref 15).

AETMS designers use the following two terms in their definition of the AETMS structure. They are referenced in this description of software and so their definitions are included here. A patch is defined as follows: "Terrain is divided into a number of patches. A patch is a small area of terrain whose altitude at each point can be determined from a polynomial. The polynomial is:

$$Z = C1+C2X+(C3+C4X)Y$$

where X,Y are the coefficients of the point and C1, C2, C3, C4 are four coefficients that vary from patch to patch, and Z is the altitude of the terrain at X,Y (Ref 4:15).

Terrain data (coefficients) from the DMA data base is stored on a disk. These concepts are described in detail by Tamburino (Ref 16).

Another important concept is that of a BAU. BAU is an ancronymn for Binary Angular Unit and equals PI/128. This is a constant used to convert azimuth into smaller units so that it is easier to access the data base.

The thesis software consists of modules that determine window bounds, check data against the bounds, and draw symbols over the terrain generated by the AETMS. The entire thesis module is titled SYMBOL. The PRSPCT executive routine call SYMBOL. Figure 3 gives a pictorial view of the

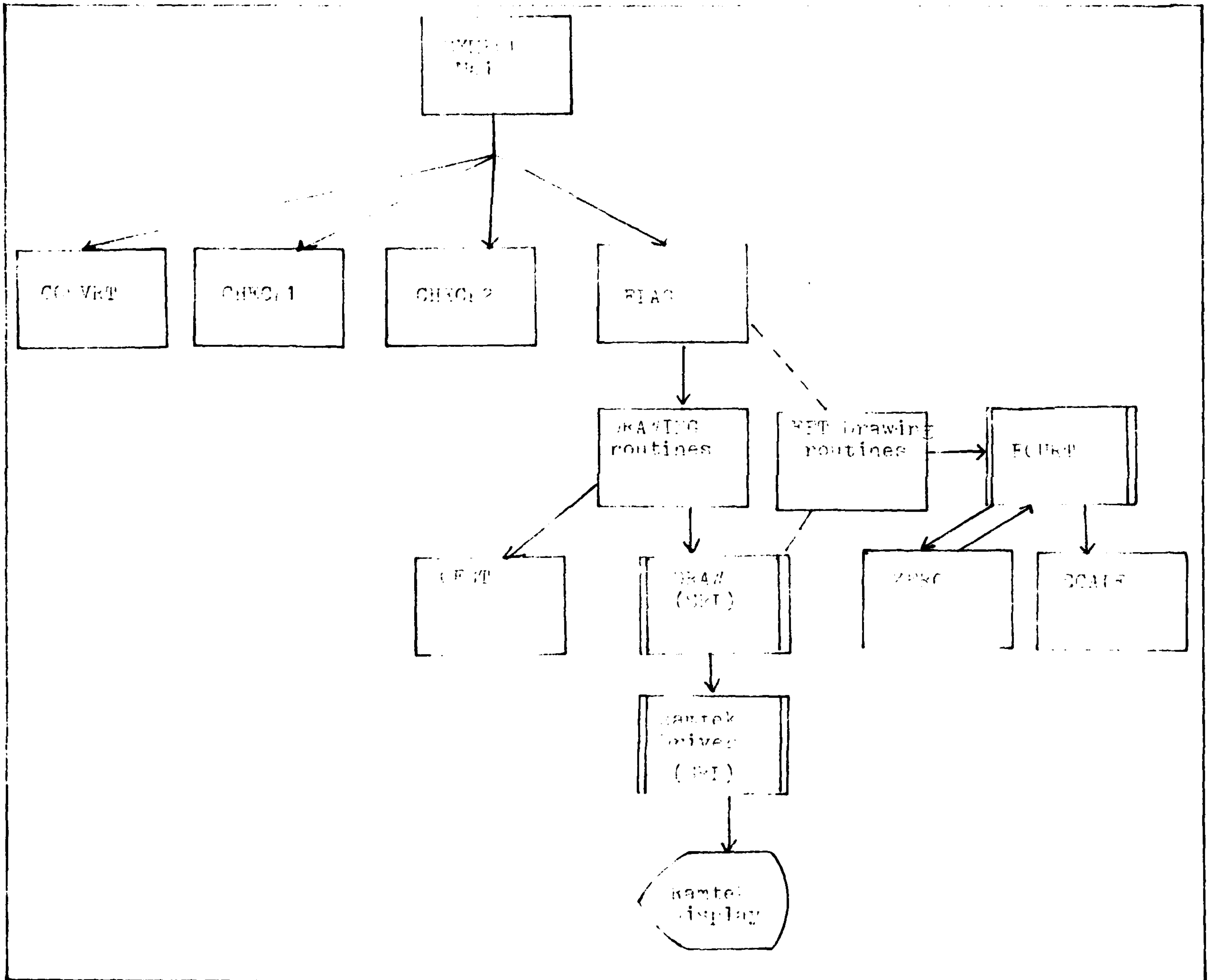
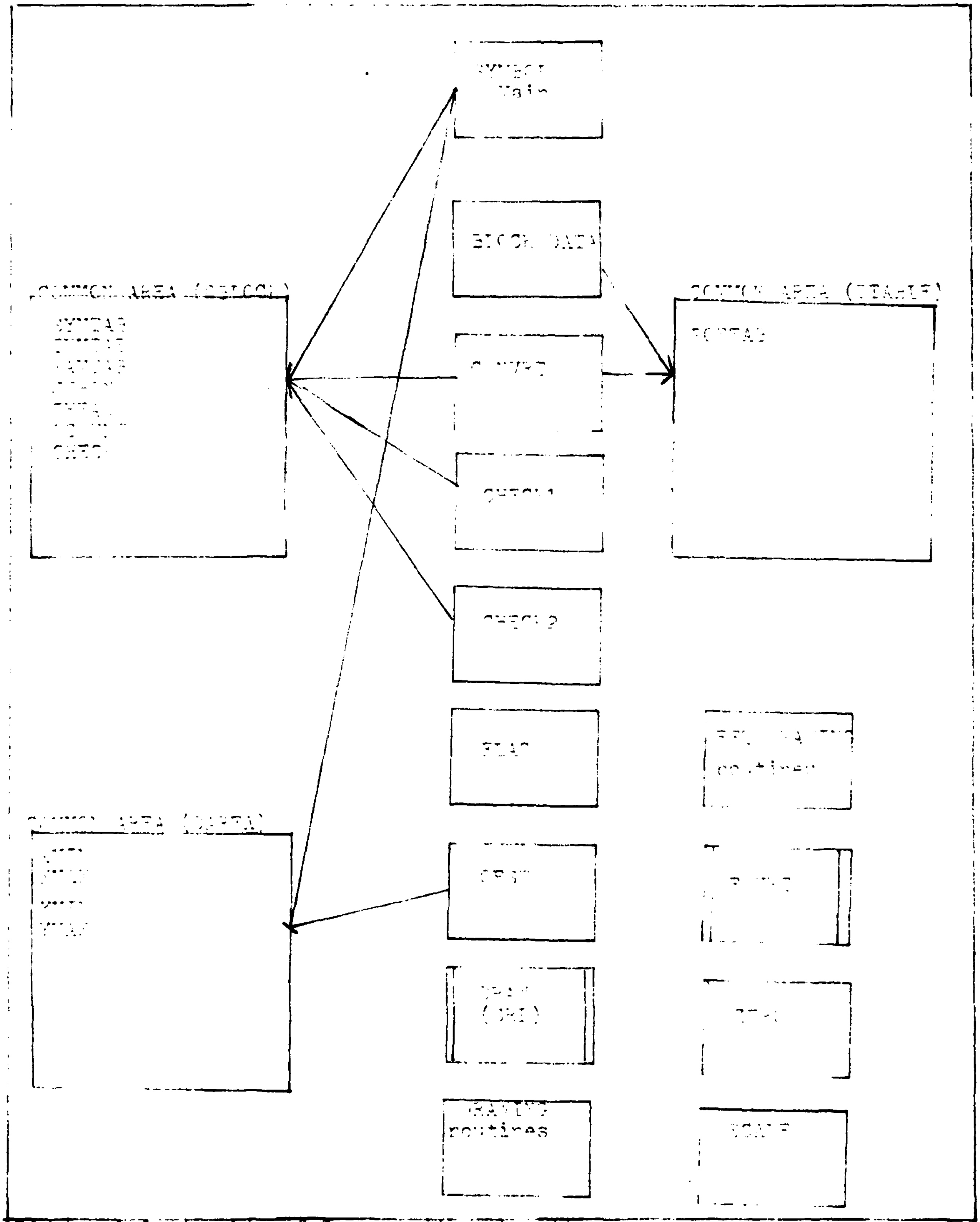


Fig. 3. Hierarchy of [illegible] with its sub-modules



2. Relationship of Common with Subroutines.

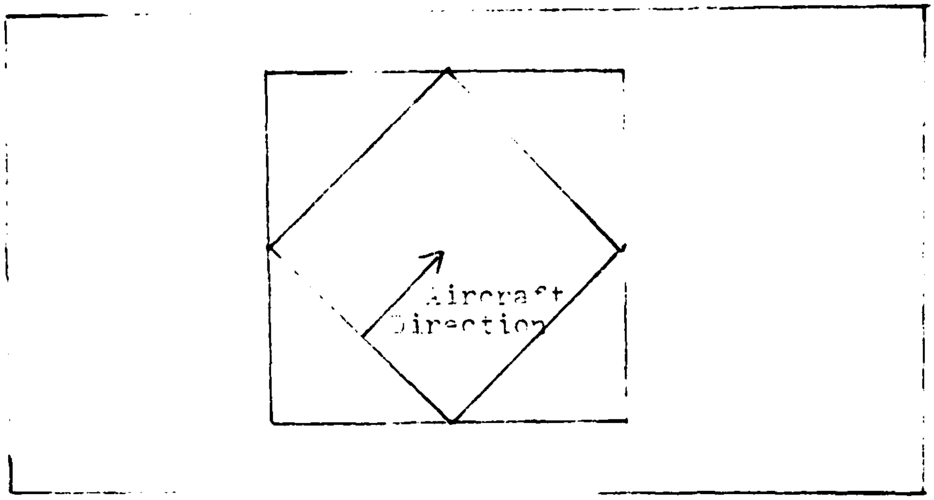


Fig 5. Rotated Display

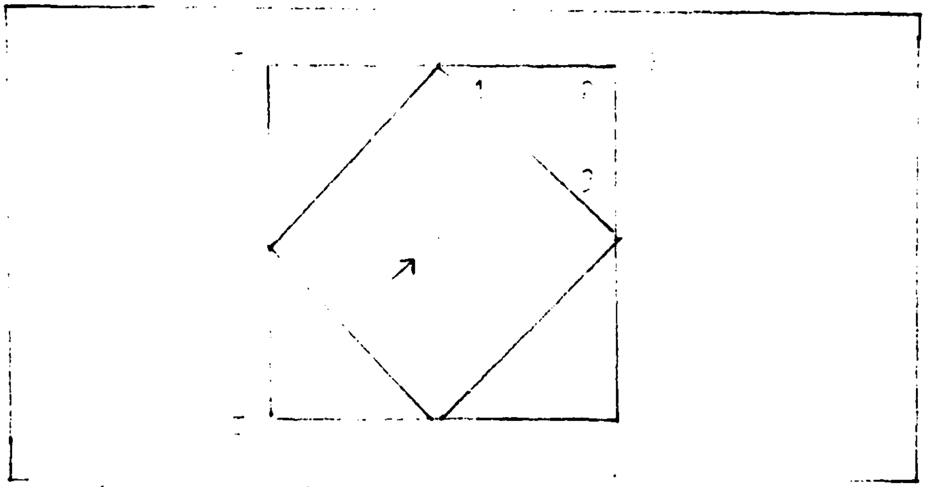


Fig 6. CHECK2 Display

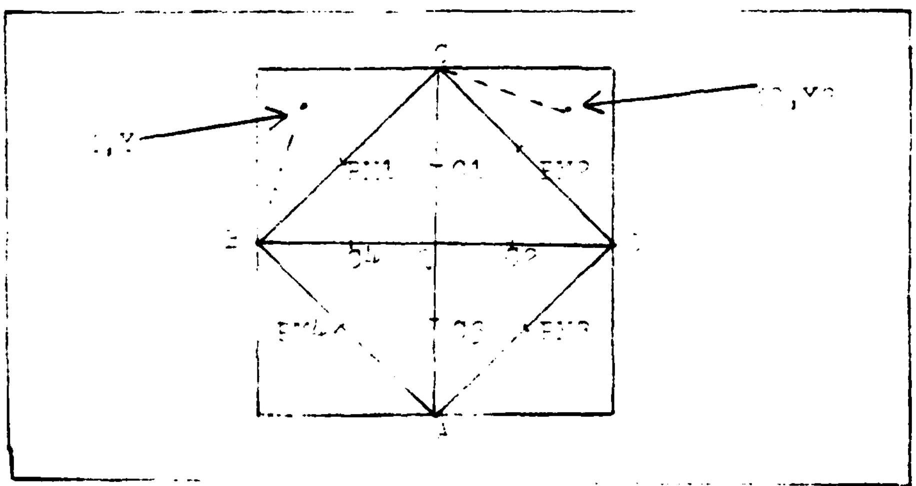


Fig 7. Rotated display with parameters

hierarchy of the modules in SYMBOL. Figure 4 gives a representation of the relationship between sub-modules in SYMBOL and the defined common areas. Appendix A contains a flowchart for the module, and a listing of SYMBOL and test results are presented in Appendix G.

SYMBOL has five main functions:

1. Determine the values of the corners of the window, aircraft heading, and conversion factors.

2. Call CONVRT each time the aircraft's position changes.

3. Call CHECK1 with the bounds found (#1 above), if and only if aircraft heading is North (0 or 360), South (180), East (270), or West (90).

4. Call CHECK2 if and only if aircraft heading is anything other than North, South, East, or West.

5. Call FLAG if and only if there are symbols within the bounds of the window.

PRSPCT passes 11 arguments to SYMBOL. Table I presents the argruments and a short definition for each.

SYMBOL uses these arguments to determine the corner values (A,B,C,D), conversion factors (POSY,POSX), and aircraft heading (IHEAD). Appendix C contains the equations for generating the needed information. Appendix B contains the variablee and table definitions for SYMBOL and PRSPCT. SYMBOL returns nothing to PRSPCT.

Subroutine CONVRT is the first subroutine called by the main body of SYMBOL. POSX and POSY, the conversion factors,

TABLE I

ARGUMENT	DEFINITION
IAY FAY	Latitude (degrees) Latitude (minutes)
IAX FAX	Longitude (degrees) Longitude (minutes)
DX/DY	Patch's angular width/length (degrees)
NX/NY	Number of Patches in memory
A	Aircraft azimuth in BAU's
D	Aircraft speed (knots)
DA	Aircraft change in azimuth in BAU's

are passed to CONVRT from SYMBOL. CONVRT uses BLOCK DATA via the common area, DTABLE. BLOCK DATA contains real world latitude and longitude in a table known as POSTAB. POSTAB also holds the flag of an object associated with a particular longitude and latitude. The dimension of POSTAB is (x,3), where the x indicates that POSTAB may contain as many symbols as may be encountered in a particular flight.

The main function of CONVRT is to convert POSTAB to display (patch) coordinates. This is accomplished by adding POSY to the latitude values, and POSX to the longitude values. The new values are then placed in SYMTAB, dimension (x,3), and returned to SYMBOL via the common area DBLOCK.

Subroutine CHECK1 is called if aircraft heading is

North (0 or 360), South (180), East (270), or West (90). The window coordinates are first sorted into minimum and maximum values and then passed to CHECK1. If the SYMTAB values fall within the bounds, then it is placed in RAMTAB. RAMTAB is later used in FLAG to tell the object drawing routines what and where to draw a symbol.

Subroutine CHECK2 is called if aircraft heading is anything other North, South, East, or West. If so, then it is known that the window is rotated as in figure 5. Checking for symbols is more complex in CHECK2 than in CHECK1. Figure 6 gives a pictorial description of the discussion to follow.

First an initial check is done against maximum and minimum values to see if the symbol is within the larger square, IJKL. Those values that are within these bounds are placed in a temporary table, TEMTAB. Next, the symbols that are within the triangles, (eg. 123), must be eliminated as they are not in the display. Midpoints and the slopes of the rotated display edges are used to do thisis checking. Figure 7 shows these points. Line slopes for lines AB, BC, CD, and AD are computed. The midpoints are necessary to determine which triangle the point is in. This is important since the next step involves a line slope comparison.

Suppose point x,y (figure 7) is a point that may or may not be in the display bounds. CHECK2 does the following:

1. Calculates which triangle x,y is in.
2. Calculates the slope of line Bx,y.

3. Slope of line BC has already been determined and is constant for the present aircraft heading.

4. Compares the slope of line Bx,y with that of line BC. If the slope of Bx,y is less than that of BC, then x,y is not transferred to RAMTAB. Consideration of another point, x2,y2, is the same except in line slope comparisoon. If the slope of Cx2,y2 is greater than that of line DC, then x2, 2 is not in the display. This is why determining which triangle the point is in is important.

Both CHECK1 and CHECK2 return a variable, LCOUNT, and a table, RAMTAB, to SYMBOL via the common area DBLOCK. LCOUNT is the number of symbols that are in RAMTAB and hence are within the display bounds. This is important in the execution of other subroutines.

Subroutine FLAG is the last routine called by SYMBOL. SYMBOL passes RAMTAB and LCOUNT to FLAG.

FLAG checks the value of the x,3 entry in RAMTAB. The value in x,3 determines which symbol must be drawn at a particular latitude and longitude.

Each object has a separate subroutine to output a representative of that object. FLAG does not return any values to SYMBOL. FLAG also sets a reset value, IFLAG, for use by the Ramtek display driver. Reset, or erasure, occurs only when a new table of RAMTAB values is passed to FLAG. After erasure, IFLAG is set to zero so that further erasures will be suppressed until FLAG is again called by SYMBOL.

The object drawing routines referred to above are

Pages 13, 14, 15 Blanks

simple end point plotting routines. FLAG passes IFLAG, RAMTAB, and a RAMTAB position pointer, to these routines.

RAMTAB latitude and longitude must be scaled to fit onto the video screen. Therefore, it is necessary to first call subroutine OFST.

RAMTAB and J are passed to OFST. OFST establishes the symbols position on the screen by converting its position in the window to the same relative position on the screen. Screen size is 480 (x) by 640 (y), with the origin n the upper left hand corner. A re-scaled RAMTAB value is returned to the calling routine.

Final output of a symbol occurs when the object drawing routine calls subroutine DRAW. This subroutine is a slightly modified version of a vector drawing routine designed and implemented by Dale Rangeler of SRL, Inc.

DRAW utilizes the Ramtek graphic display driver to output vectors. Each drawing routine passes a 4X4 matrix and IFLAG to draw. DRAW simply writes a line between the beginning x and y coordinates and the ending x and y coor dinates. The number of calls to DRAW by a particular object drawing routine is a function of the number of vectors that an object is composed of.

The program to output Fourier transforms of symbols is essentially the same except for the addition of three subroutines.

Subroutine FOURT is a program by Norman Brenner from the basic program by Charles Rader. It uses the Cooley-

Tukey Fast Fourier Transform and is written in USASI basic fortran (Ref 4). This subroutine is called by the drawing routines to determine the fast Fourier transform data points of the object in question.

Subroutine ZERO zeroes out data after the third harmonic of Fourier data. This is necessary to get a usable representation of the object from the data.

Subroutine SCALE reduces the data points to a size that is compatible to the Ramtek display dimensions.

Testing SYMBOL consists of checking values on the bounds and corners, as well as values in between the bounds. Appendix F contains a SYMBOL user guide. Appendix G contains a program listing and test values.

III. LITERATURE REVIEW

This chapter has a twofold purpose. The first objective is to present some ideas about contemporary map displays and information requirements, and to show that the AETMS is a versatile system that does infact, fulfill these information requirements.

The second objective is to put forth some ideas about research that has been done concerning symbols, their size and color, and to give a short synopsis of other problems that are associated with visual displays.

Human factors engineering evaluation has become an important pre-design consideration. This is necessary because a tool that is designed for human use reduces mistakes and frustration. In light of this, the AAAT-1 Group tasked the Aerospace Medical Research Laboratory (AMRL) to do an exhaustive search of literature on current map systems. AMRL found that there are six map systems in existence today. The following summary of map types describes each map and its major limitations, and is described in more detail in a report prepared for the AAAT-1 Group by AMRL (Ref 10).

The most primitive map is a handheld chart. It has the obvious disadvantage of being nearly impossible to use while flying at high speeds and low altitudes. Currently, no fighter system uses hand held charts as a map system.

Direct view maps are paper maps mounted between rollers. The map moves as does the airplane. Movement is a result of computer or doppler input. The major disadvantage of this system is that it is difficult to imput steering or heading changes into the system.

The projected map system uses a rear-project microfilmed transparency of the original map. It is limited because it is costly and cannot be changed for a specific flight.

A combined map/CRT is a system that displays a map on a CRT. Researchers have found that there are legibility problems with this system.

Electronic displays generate all information electronically. Currently, there are no systems with airborne capablities for using an electronic system. Designers are hesitant to use this type of a system because it is easily detectable by enemy radar.

Terrain avoidance and terrain following (T/A, T/F) systems allow pilots the ability to hide their aircraft under conventional electronic det on systems by allowing them to hug the terrain. The AETMS is a system designed to be a T/F system. "The AETMS is more than just a horizonital or vertical situation map system. It is an integrated information system that will supplement current aircraft systems, giving the pilot the capability to negotiate low level, high speed profiles" (Ref 50:10).

Information parameters for future aircraft systems are demanding. They were described by Crawford (Ref 1) and are listed below:

1. Altitude below 500 ft.
2. Speeds proportionate with pilot and aircraft capabilities.
3. Sensing equipment that is 'invisible', i.e., not seen by enemy radar.
4. Use of missiles which require accurate positioning.
5. Minimizing detection time by maximizing time 'under' radar.

The AETMS was designed to provide answers to all of these parameters. The pilot will be able to fly below 500 ft at his maximum speed. It is an internal system and hence, it does not leave a signature that can be read by enemy detection systems. It is a highly accurate system since it is an exact copy of digitized terrain data. The pilot can stay below radar until just before weapons delivery. This is because he will not need to manually search for targets. Target points will be displayed on the screen. The system has the advantage of providing a look ahead feature for the pilot. Look ahead gives the pilot time to plan at alternate routes or targets. It is obvious that this system can reduce pilot stress, and therefore, will help prevent errors from fatigue.

"The Joint Tatical Information Display Syatem (JTIDS) is a digital, secure, jam-resistant, communication system for a real-time command and control of combat operations"

(Ref 1:2). JTIDS monitors have the responsibility for standardization of a symbol set for aircraft systems. This is necessary because contractors develop a set of symbols each time they bring a new aircraft system on-line. A Symbology Standardization Committee (SSC) was formed to develop the standards. The SSC made several recommendations, a few of which are important to this discussion. A more detailed explanation of the JTIDS, SSC, and its recommendations can be found in a report prepared by the SSC (Ref 2).

The important recommendations are:

1. Largest dimension of a symbol should not be greater than 17 minutes of visual angle.

2. The symbol height to width ratio should be 3:2 (Ref 2).

Past experiments have shown that 17 minutes of arc is optimum. The following equation is used to calculate the size of a character:

$$H=2DTAN(\Theta/2*60)$$

D= viewing distance in inches from eye to surface
Θ= minutes of visual angle (arc)
H= symbol height in inches (Ref 2)

A study by the McDonnell Aircraft Company (McAir) presents a definition of a test JTIDS symbol set in the F-15 system. They established the same basic principles on object size as did the SSC. In addition, they determined that color is an excellent tool to use in preventing display clutter. Clutter is a result of trying to place too much information on a display. The operator becomes overloaded with informtion and therefore, may miss important points. McAir used the standard three color system: red indicates hostility or danger, green represents friendliness or safety, and yellow indicates an unknown situation (Ref 12).

Electronic display systems also present other complications in developing an optimum visual display. These problems include flash or flicker, excess brightness, ambient light, and the adaptation differences required for night vision.

Flash or flicker is a phenomenon typical of electronic displays. The image on the screen appears to flicker. This is especially noticeable in a dark environment. It increases fatigue and eye strain. Flicker can be minimized by reducing contrast and background brightness.

Excess brightness is another problem associated with electronic displays. Either a symbol or its background is too bright. Both of these possibilities increase the chance that other elements of the display will be hard to distinguish. This problem also causes fatigue and eye strain.

Ambient light is the light that surrounds the display. It changes some of the other parameters that must be considered in developing good displays. It may decrease contrast or foreground brightness depending on its intensity. These changes may in turn increase the chance of eye strain or fatigue.

The adaptation requirement necessary for a change from day to night vision must be considered when designing a visual display system. It is more difficult to see at night. Looking out into the dark and then down into a lighted screen causes immediate adaptation problems. This is probably the hardest problem to solve or minimize, and can potentially be one of the most dangerous dilemmas that a pilot may face.

The preceding discussion was a summary of a few of the problems that others have observed and have tried to solve. This thesis does not consider any of these problems, but the experiment was designed with them in mind. A detailed description of these problems and their solutions can be found in a work by Ketchel and Jenney (Ref 8) and Soliday (Ref 14).

IV. SYMBOL AND COLOR SET

This chapter will give some insight as to why a set of symbols and colors were chosen for the experiment.

The object set for this experiment consists of eight symbols and their Fourier transforms (FFT). Appendix D contains a pictorial representation of the symbols as they are seen on the Ramtek display.

The objects in the symbol set represent cultural and linear objects, and come from three categories. These catagories include obstructions, targets, and linear objects. Table II shows the categories and objects.

The obstruction set is mandatory since the AETMS is designed to be used in aircraft flying at low altitude and high speed. The pilot needs to be pre-warned of any upcoming obstructions in order to avoid them.

Landmarks and target categories are necessary for similar reasons. The pilot needs a look ahead capability in order to prepare for weapons delivery or a change in course.

Linear objects will be most difficult to manually identify, because of high speed and a perspective background. Hence, linear objects are included in the symbol set.

Kabrisky hypothesized that a two dimensional Fourier transform occurs on visual input, and that this data is used by the human visual system for rocessing (Ref 7). Therefore, the Fourier transforms are included in the set so

TABLE II

CATEGORY	OBJECT
Obstructions (Landmarks)	Bridge Tall Building Electrical Wires
Targets	Target Designator Pointer
Linear (Landmarks)	Landing Strip Highway

that it can be determined through testing whether a simple representation of a object or its FFT is easier to identify. Adding FFT's to this thesis is for testing purposes only, the final symbol set will not include FFT's.

Color is a tool that can help enhance the usefulness of any display. Color sparks curiosity. A color coded system can also help eliminate clutter. It is important to find the best combination of colors for the AETMS. Best can be described as any number of things, including appeal, functionality, and legibility.

Red, green, and yellow are traditional aircraft instrumentation colors. Red indicates hostility or danger, green shows friendliness or safety, and yellow symbolizes an unknown condition. Table III shows foreground and

TABLE III

SYMBOL COLOR	BACKGROUND COLOR	SIGNIFICANCE
Red	Grey	Hostile
Green	Grey	Friendly
Yellow	Grey	Unknown
Grey	Red	Hostile
Grey	Green	Friendly
Grey	Yellow	Unknown
White	Black	Friendly
Black	White	Hostile

background color combinations.

Determining a good combination of colors and symbols will help make the AETMS a versatile and functional aid to pilots.

V. METHODOLOGY

This chapter reviews how the experiment was to be conducted, and explains what video output is necessary for such an experiment. The AAAT-1 Group has the equipment necessary to make a video recording of output produced by the AETMS and SYMBOL. It is possible therefore, to produce a sequence of AETMS and symbol overlay outputs that can be used in an experiment.

The output should consist of three separate tasks: the learning task, a non-stress identification task, and a stress producing validation task.

Task 1, the learning task, is necessary because past results have shown that subjects involved in tests such as this, need to learn about the task before they can perform it accurately (Ref 3). If this segment is ignored, then the results may not be valid and conclusions should not be drawn. This phase consists of three events. First, the subject is shown what terrain output looks like. Secondly, the subject is shown simple symbols and their Fourier transforms on a plain background. Finally, the subject is presented the symbols overlayed onto the terrain.

Task 2, the non-stress producing identification, shows a simple sequence of simple objects, their Fourier transforms, and a sequence of colors to the subject. The subject views the symbol and indicates what he thinks it represents. The subject's identification is either corrected or validated. The same interaction occurs in color identification.

Task 3, the stress producing validation segment, is done to validate that the symbols really do represent some identifiable object to the subject. Symbols are presented in a random order and the subject must identify it and indicate whether or not the object is a threat.

The experiment is conducted in the same order as the video product described above.

Phase 1 corresponds to Task 1. During this phase, the subject becomes oriented to what the terrain and symbols should look like. It is also during this phase that the subject and the one conducting the experiment develop a rapport. The subject must feel at ease during the experiment, and this phase allows this to happen.

Phase 2 is implemented by using Task 2. The subject and experimenter interact during this phase. Phase 2 events occur in the following order: First, a symbol appears on the screen; secondly the subject indicates what he thinks the symbol represents; finally, his answer is either corrected or validated. The same sequence of occurs for color identification.

Finally, Phase 3 is conducted using output from Task 3. Symbols are generated by the task. As soon as the subject notices the symbol, he must identify it, and indicate whether it is a threat, a weapons delivery point, or neither of these. This phase is used to validate the results of previous phases.

The background data and instructions for the experiment are presented in Appendix E, along with the experiment questionaire, experiment tally sheet, and post-experiment questionaire.

VI. CONCLUSIONS

The AETMS will be a powerful tool once a few logistics problems are overcome. The database for the map is quite large and requires a great deal of memory. It has been suggested that bubble memory be used to solve the space and weight problems associated with other memory devices. In addition, bubble memory has a faster access time than other forms of memory.

Furthermore, additions of the symbol overlay and a threat overlay being developed by the University of Dayton, will make the entire package a versatile tool for aircraft systems.

VII. RECOMMENDATIONS

It is possible to task SYMBOL with PRSPCT and the data base by using the overlay task building system available on the PDP 11/45. The thesis software can be reduced to an overlay of itself. There are at least two ways to do this One way is to overlay every subroutine onto the main body of SYMBOL. Another way to do it, is to break the software into logical groups, such as: SYMBOL (SYMBOL, CHECK1, CHECK2, CONVRT), FLAG (FLAG, DRAW, OFST, BLOCK DATA), and PICTURE (all of the symbol drawing routines). The Fourier transforms programs need to have one more module, FFT (FOURT,SCALE,ZERO). These groups can then be overlayed onto PRSPCT.

It becomes hard to make a large latitude and longitude table for symbols since memory is a problem. Therefore, a more compact table should be designed to alleviate the problem. One way to do this, is to divide BLOCK DATA into subsets that can be overlayed onto each other. There would be three sections of block data, one for a latitude table, a second for a longitude table, and a third for a flag table.

A subroutine needs to be added to check the aircraft's altitude. PRSPCT monitors altitude and so this parameter could be passed to SYMBOL. Symbol color can then be changed as the altitude increases or decreases.

The symbol set must be tested to determine which symbols should be included in a master set. More symbols need to be added to the current table so that testing can be conducted to determine that master set.

Finally, the entire system should be documented so that those that do not have an intimate knowledge of the system can use it.

Bibliography

1. Carel, W.L., et al. "Design Criteria for Airborne Map Displays." Culver City, California: Hughes Aircraft Company, March 1974.

2. Cassella, J.R., et al. "Joint Tatical Information Distribution System (JTIDS), Standard Symbology Phase I, Cockpit Display Presentation Formats and Flight Test Symbol Set." ASD-TR-77-31, Wright-Patterson Air Force Base, Ohio: Aeronautical Systems Division, June 1977.

3. Chapanis, Alphonse. Research Techniques in Human Engineering. Baltimore: Johns Hopkins Press, 1959.

4. Cooley, J.W. and J.W. Tukey. "An Algorithm for Machine calculation of Complex Fourier Series." Mathematics of Computation, 19: 297-301 (April 1965).

5. Crawford, N.W. "Low Level Attack of Armored Targets." Santa Monica, California: The Rand Corporation, August 1977.

6. Dreyfuss, Henry. Symbol Sourcebook. New York City: McGraw-Hill, 1972.

7. Kabrisky, Matthew. A Proposed Model for Visual Information Processing in the Human Brain. Urbana, Illinois: University of Illinois Press, 1966.

8. Ketchel, J.M. and L.L. Jenney. "Electronic and Optically Generated Aircraft Displays: A Study of Standardizaton Requirements." Washington, D.C.: Office of Naval Research, May 1968.

9. Kopala, C.J. "The Use of Color-Coded Symbols in a Highly Dense Situation Display." Wright-Patterson Air Force Base, Ohio: Air Force Flight Dynamics Laboratory, March, 1980.

10. Kuperman, G.G. and A.J. Defrances. "Airborne Electronic Terrain Mapping System: A Literature Review, AMRL-TR-79-92, Wright-Patterson Air Force Base, Ohio: Aerospace Medical Research Laboratory, October, 1979.

11. Kuperman, G.G., et al. "Information Requirements for Airborne Electronic Maps." IEEE Transactions from NAECON 1980, 1046-1051 (April 1980).

12. McDonnell Aircraft Company, "F-15 JTIDS Simulation." St. Louis, Mo: 1979.

13. Sander, D. E. Personal Communication.

14. Soliday, S.M. and J.R. Milligan. "Simulation of Low Altitude High Speed Mission Performance, Vol. II, Effectiveness of a HUD Display for Take-off and Landing in Fighter Aircraft." SEG-TR-66-67, Wright-Patterson Air Force Base, Ohio: Systems Engineering Group, February 1967.

15. Systran Corporation. "The Airborne Electrnic Electronic Terrain Mapping System Documention." Dayton, Ohio: Systran Corporation, 1978.

16. Tamburino, L.A. "Airborne Electronic Terrain Map Display System, AFAL-TR-77-232, Wright-Patterson Air Force Base, Ohio: Air Force Avionics Laboratory, December, 1977.

APPENDIX A

FLOWCHARTS

This appendix contains the flowcharts for SYMBOL and its sub-modules. Note the difference in figure 10 and figure's 11 and 12. Figure 10 represents a flow chart of data that has not been transformed to Fourier data. Figure's 11 and 12 represent the Fourier transform modules.

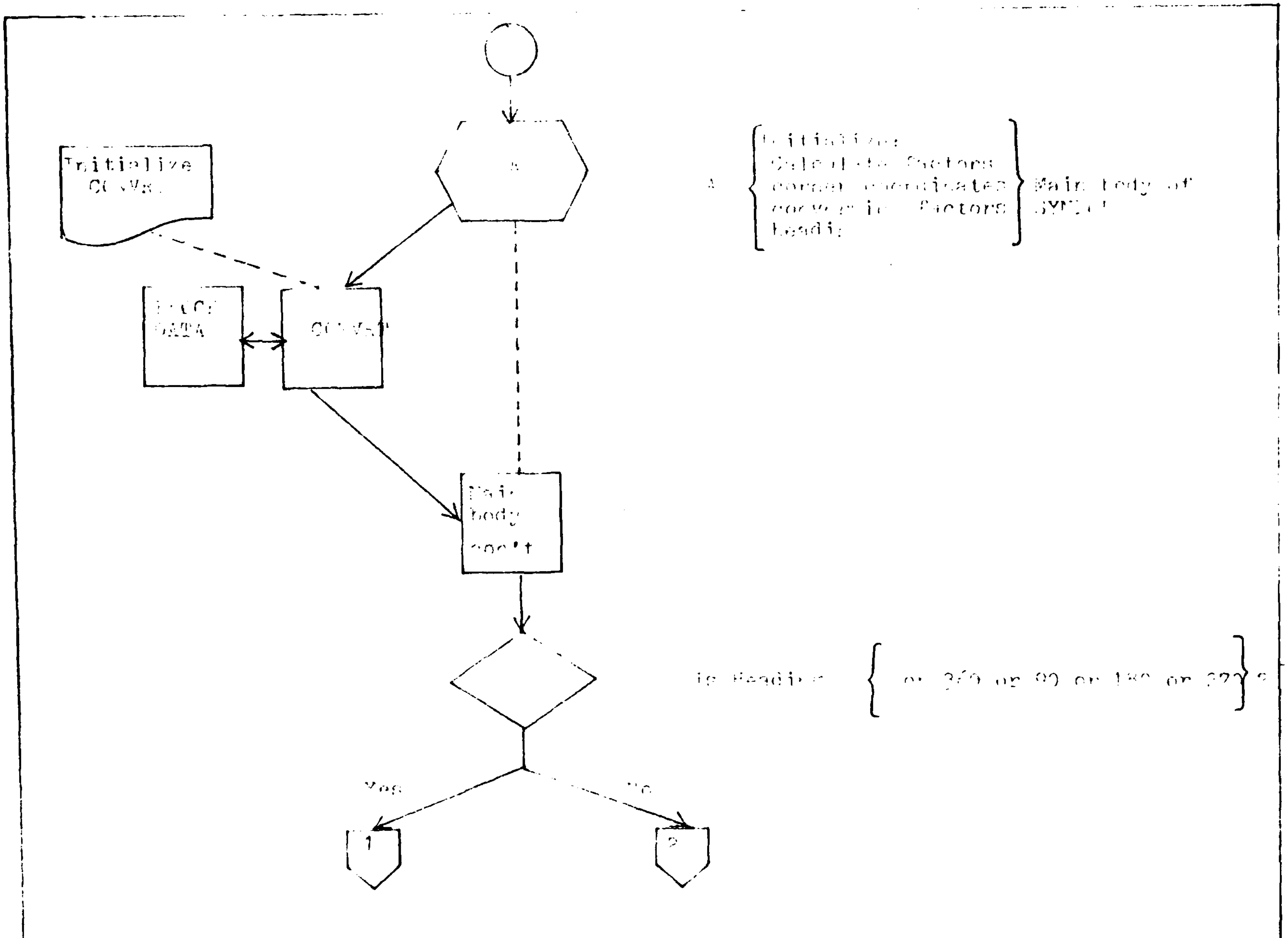


Fig 2. Flow chart

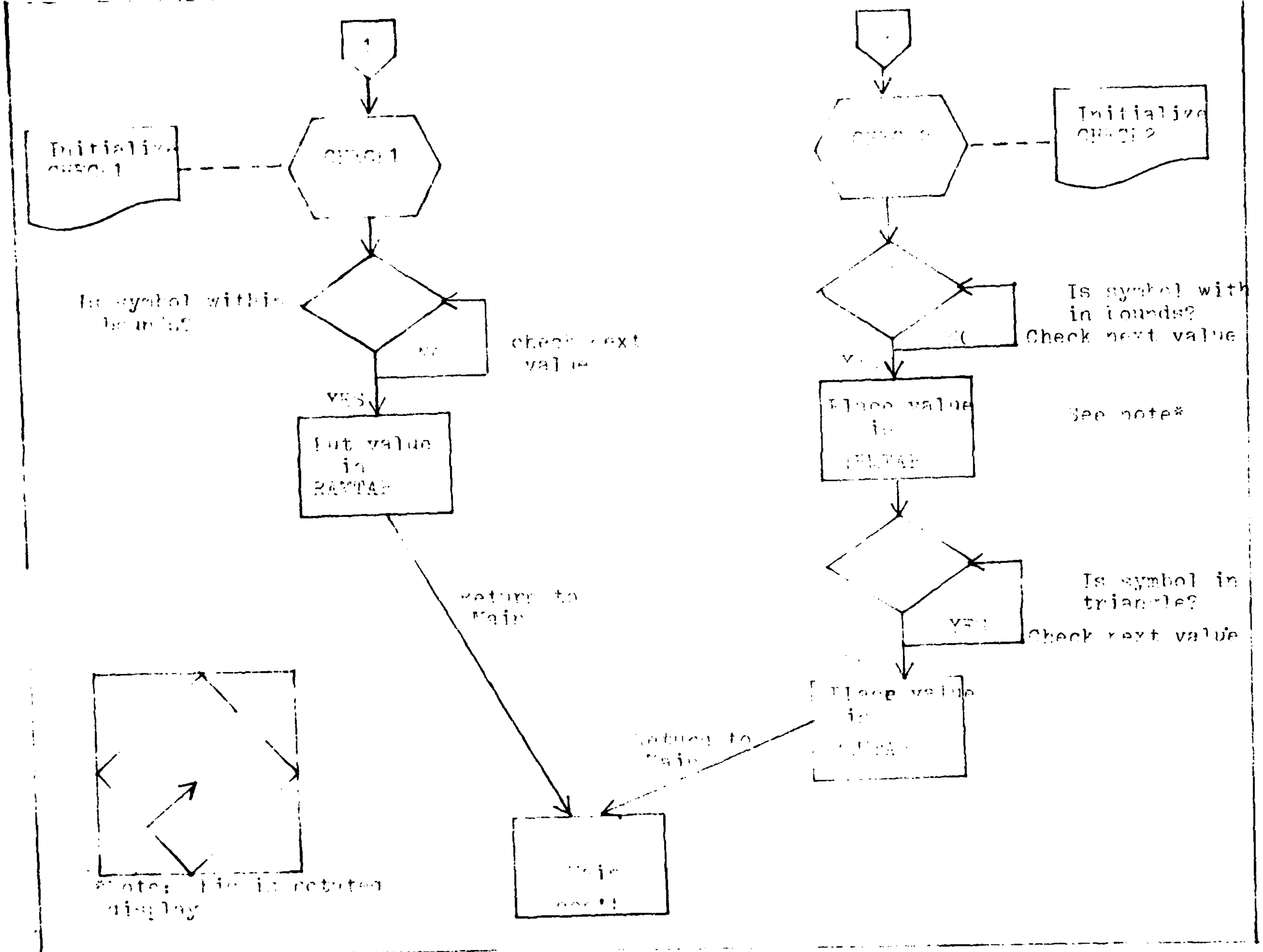


Fig. 2. Flow Chart

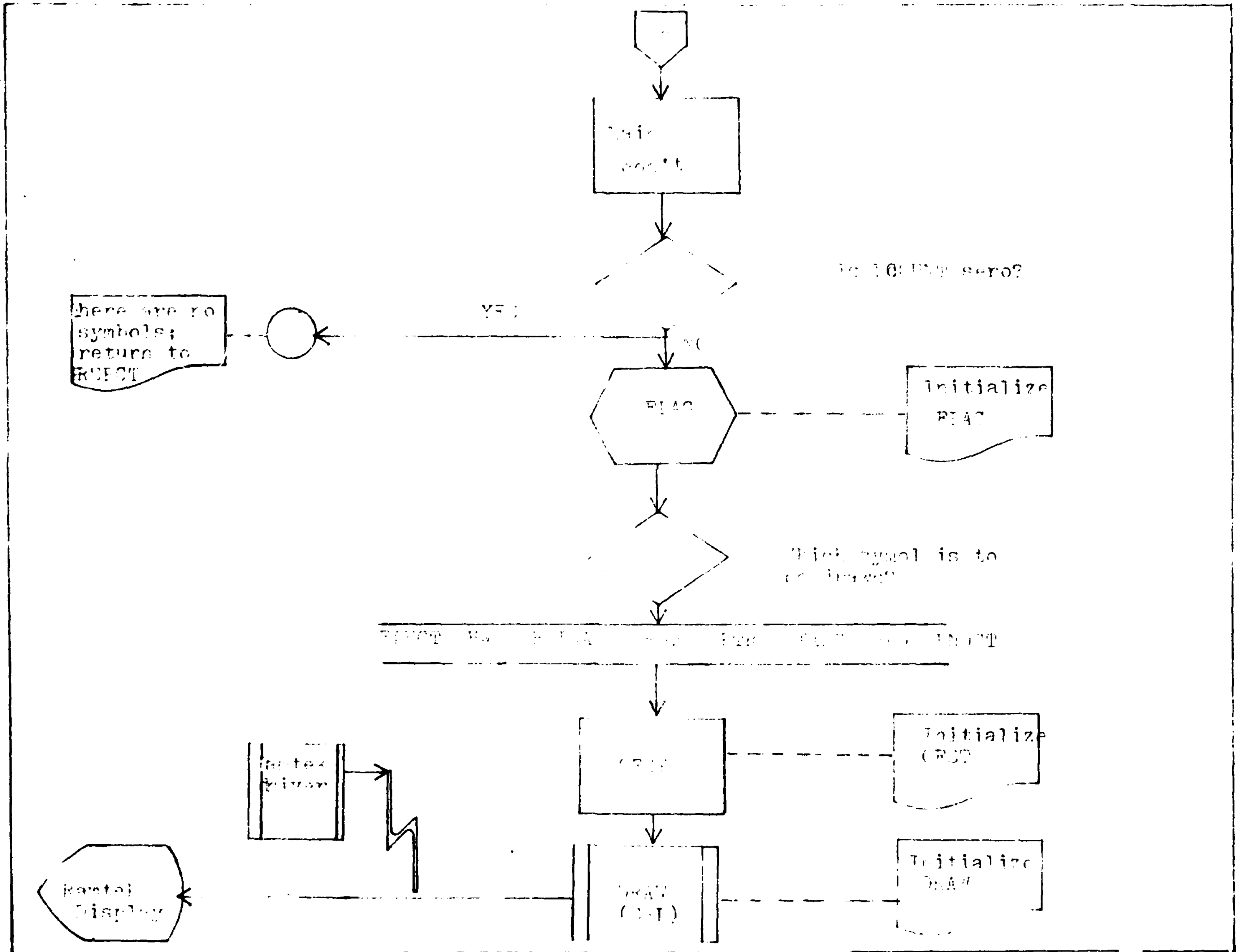


Fig. 10. Flow chart

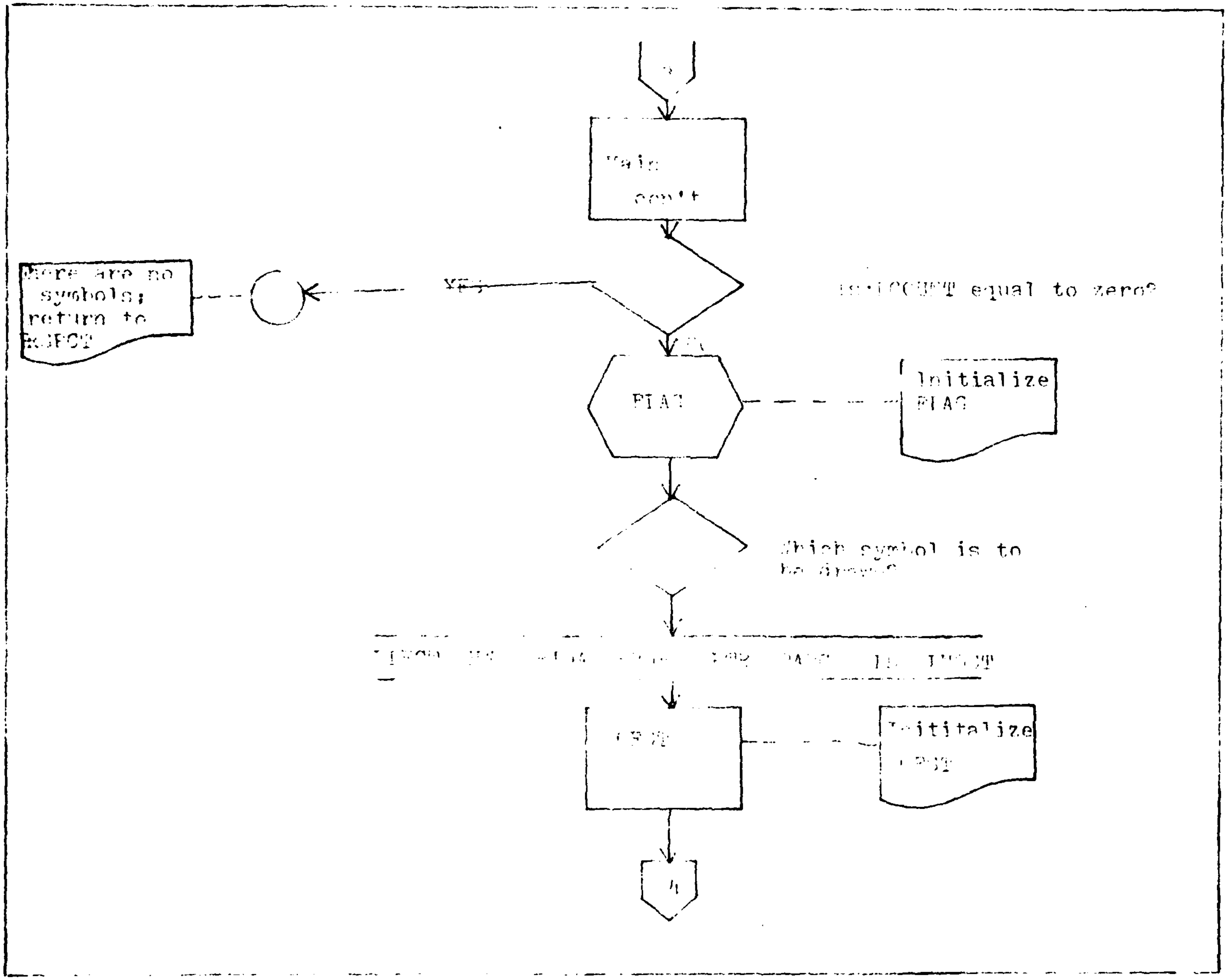


Fig. 11. Flow chart

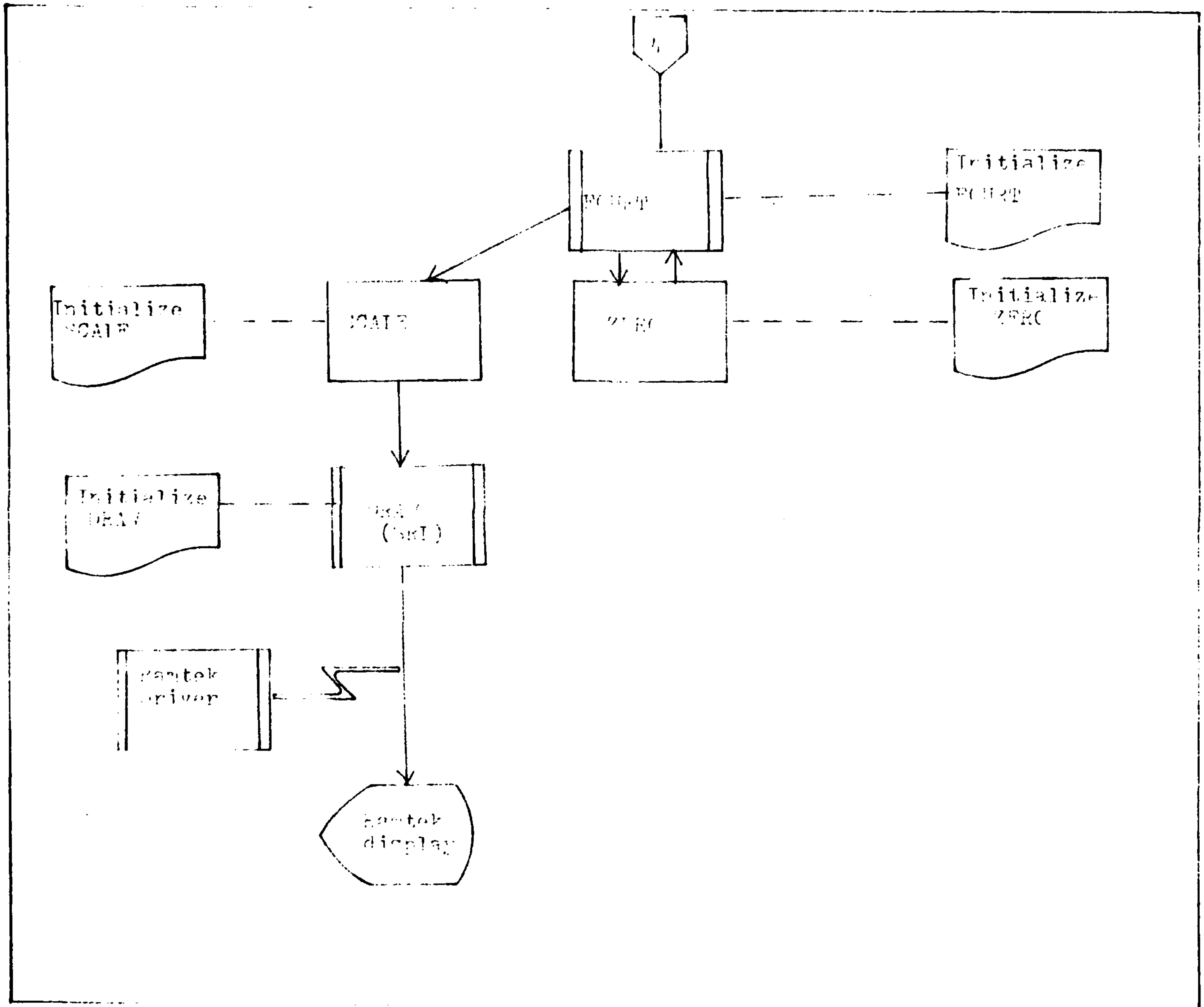


Fig 12. Flow Chart

APPENDIX B

VARIABLE AND TABLE DEFINITIONS

This appendix contains the variable and table definitions necessary to understand their use in SYMBOL and PRSPCT. Table IV describes the AETMS defined variables; Table V describes variables defined in SYMBOL; and Table VI describes the tables used in SYMBOL. Common areas are described in Table VII.

TABLE IV.

NAME	DESCRIPTION
BAU	$\pi/128$
B	1 BAU
A	aircraft azimuth
Patch	a digital description of a small area of terrain in memory
DH	display horizon (5.52)
DA	change in azimuth over one time slice
D	aircraft speed in knots
IWY IWX IX IY JX JY KX KY	Corner coordinates of the window

TABLE V.

NAME	DESCRIPTION
POSX POSY	Coordinate conversion factors
CONS CONC	sine and cosine factors
AX,AY BX,BY CX,CY DPX,DPY	Corner coordinates (window)
X;YMIN X;YMAX	boundary values
SAB SBC SAD SDC	Edge (window) slopes
LCOUNT	Number of symbols in the window

TABLE VI.

NAME	DESCRIPTION
RAMTAB	Table of values within the bounds
SYMTAB	Latitude and Longitude values in patch coordinates
TEMTAB	Temporary table
POSTAB	Latitude and Longitude values in real world coordinates

TABLE VII.

NAME	OWNERS
DTABLE	BLOCK DATA, CONVRT
DAREA	SYMBOL, OFST
DBLOCK	SYMBOL, CHECK1, CHECK2, CONVRT

APPENDIX C

EQUATIONS

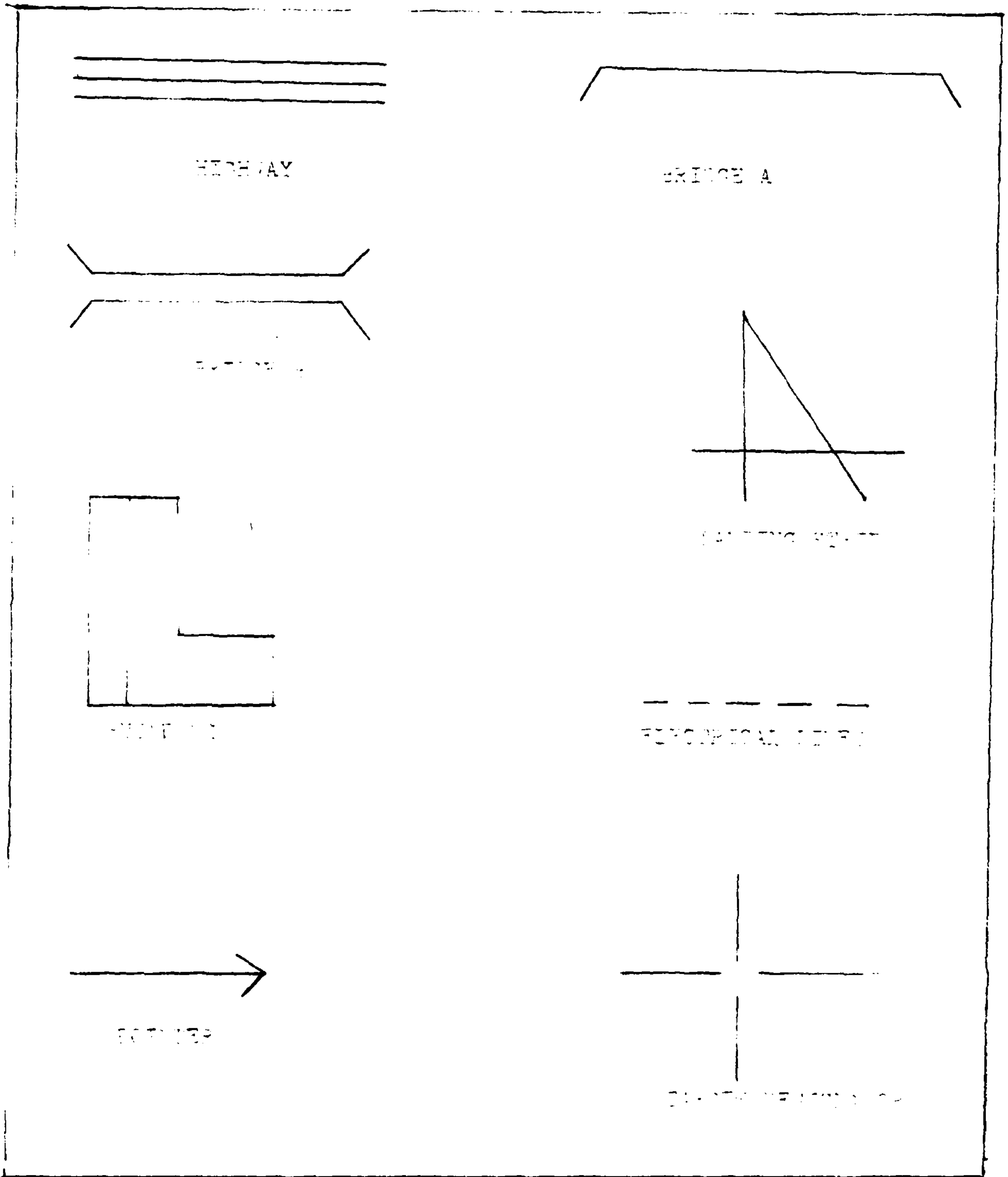
This appendix contains the equations necessary to determine the conversion factors in symbol. All of these equations were developed by AETMS designers (Ref 15).

```
IAY,FAY = (IAY,FAY)+D(DA*(B/2)*SINA+COSA)/DY
IAX,FAX = (IAX,FAX)+D(SINA-DA*(B/2)*COSA)/DX
   POSX = D(SINA-DA*(B/2)*COSA)/DX
   POSY = D(DA*(B/2)*SINA+COSA)/DY
   CONS = ((DH*SINA)/(2*DX))
   CONC = ((DH*COSA)/(2*DY))
```

APPENDIX D

SYMBOL SET

This appendix contains the eight symbols that can be displayed by the thesis software. The symbols are international symbols compiled by Dreyfuss (Ref 6).



[illegible] 13. Symbols

APPENDIX E

EXPERIMENT INSTRUCTIONS AND DATA COLLECTION FORMS

This appendix contains the data collection forms necessary to compile dataB from the experiment. Included in this appendix are the background and instructions, an experiment questionaire, an experiment tally sheet, and post experiment questionaire.

Instructions and the background information are an important part of any experiment. If the subject is not interested in the experiment in which he will participate, then he may become bored or inattentive. Therefore, the subject is first given some background information about the AETMS and why the experiment is to be conducted. Furthermore, if the subject does not understand the instructions, he will not be able to complete his task and results may be invalid (Ref 3).

INSTRUCTIONS

This is a three part experiment. Part 1 is an orientation phase. Part 2 is an identification phase and Part 3 is a validation phase. I will give you some background on the AETMS and instructions before we begin. Are there any initial questions?

Part 1 is an orientation phase. You will be shown a sample terrain, symbols on a plain background, and symbols overlayed on the terrain. Your task is to view the output so that you will have an idea of the visual output that will occur in the other two phases. Feel free to ask questions during this phase.

TASK 1 is presented to the subject.

Part 2 is an identification phase. We will interact in a question and answer format. You will be presented a symbol on the screen. Tell me what you think it is. I will validate your answer if you are correct, and correct your answer if it is incorrect. We go through the same sequence of events with different color combinations. Are there any questions?

TASK 2 is presented to the subject.

Part 3 is a validation phase. Symbols will be randomly presented. Once you see a symbol indicate what it is. Also indicate whether or not it is hostile or threatening. This is done by color coding. Colors and symbols seen in this phase are the same as seen in the previous phases.

TASK 3 is presented to the subject.

The experiment is now over. Please answer the questions on the questionaire.

EXPERIMENT QUESTIONAIRE

1. Task 2: What does the symbol represent?
2. Task 2: What does the color represent?
3. Any task: What comments do you have about this phase?

EXPERIMENT TALLY SHEET

TASK 2

OBJECT PLACE A 1 IF THE OBJECT/COLOR WAS CORRECTLY IDENTIFIED

Highway
Electrical lines
Bridge a
Bridge b
Pointer
Target
Landing strip
Building

COLOR

red
green
yellow
black
white

TASK 3

OBJECT COLOR COLOR/correct/SYMBOL (1 indicates correct)

Building green
Target red
Pointer red
Electrical lines yellow
Highway green
Landing strip green
Electrical lines red
Bridge A yellow
Target red
Bridge B yellow
Building black
Pointer white

POST EXPERIMENT QUESTIONAIRE

1. Did you feel at ease during the experiment? If not, why not?

2. Were there too many distractions during the experiment?

3. Were the instructions clear and was there ample time to ask questions?

4. Rate the display with these factors. 1 indicates a high rating, 5 indicates a low rating.

Legibility 1 2 3 4 5

Understandibility 1 2 3 4 5

Functionality 1 2 3 4 5

Appeal 1 2 3 4 5

5. If you were in the position of accepting this display system of a SPO would you

a. Accept the system as you saw it in the experiment?

Yes No (if yes, goto 6)

b. Require a change to the terrain output? Yes No

c. Require a change to the symbols in the symbol set?

Yes No

What changes would you expect?

6. Imagine that you are a pilot tasked with destroying an enemy oil field. Your mission will take place at night and the weather leaves much to be desired. You therefore must depend on the display system in your aircraft. Will you feel at ease in trusting your life to this system?

Why?

7. Please feel free to make any other comments about the display or experiment.

APPENDIX F

USER'S GUIDE

This appendix contains a guide for those who would like to use the thesis software.

The following short program can be added to call SYMBOL.

```
Program Runner
Print*,'input the information that will simulate AETMS inputs'
Read*,IAY,IAX,FAY,FAX,A,DX,DY,NX,NY,D,DA
Call SYMBOL(IAY,IAX,FAY,FAX,A,DX,DY,NX,NY,D,DA)
END
```

Figure 14 represents BLOCK DATA as it is in the thesis software. The first 21 entries represent IAX,FAX and the second 21 represent IAY,FAY. The last 21 entries represent flag values. The following equation will break each table entry to a minutes and degrees entry. Input into RUNNER must be in minutes and degrees.

Let the entry we are to consider be the first one. IAX,FAX= 15.5 Break down is done in the following manner: IAX=15 (integer); FAX=.5*60 (real) the IAY that corresponds to 15.5 is 10.75. Break down is similar to the IAX,FAX break down. IAY=10 (integer) and FAY=.75*60 (real). Input for these two values is: IAY IAX FAY FAX DX DY A NX NY D DA 10 15 30. 30. 10. 10. 0. 64 64 0. 1.

All these values are defined in APPENDIX B. A and DA must be input as BAU values. Conversion from degrees to BAU's is done with the following equation:

(azimuth in degrees)*(PI/128)

```
DATA POSTAB/15.5,25.5,35.5,45.5,55.5,65.5,75.5,
*85.50,86.79,87.00,83.12,83.59,84.10,83.9,83.51,84.26,
*85.69,85.22,86.13,86.61,85.0,10.75,20.75,30.75,40.75,50.75,
*60.76,70.75,80.75,35.22,35.39,35.59,35.45,35.16,34.21,34.0,
*34.26,34.59,34.16,34.76,34.61,35.0,20.0,30.0,40.0,50.0,
*60.0,70.0,80.0,90.0,90.0,30.0,20.0,20.0,20.0,20.0,20.0,
*20.0,20.0,20.0,20.0,20.0,20.0/
```

Figure 14. Block Data

Similar conversion for the other values in BLOCK DATA will allow the user to the ability to retrieve all symbols.

APPENDIX G

SOFTWARE LISTING

This appendix contains a listing of all the modules in SYMBOL and the test data. Note that there are three subroutines named FLAG. This was necessary due to the fact that there is only 28K of usable memory in the PDP 11/45. The program could not be compiled as one unit. Therefore, there are three separate tasks. TASK I is the entire module with eight symbol drawing routines. TASK II uses the same SYMBOL, CHECKI,CHECKII, OFST, and CONVRT that TASK I uses. TASK II contains subroutines that output four of the symbols as Fourier data, while TASK III outputs the other four. FLAG is varied depending on which task it is a part of.

Also contained in this listing of the executive routine in the AETMS system, PRSPCT. SYMBOL is called by PRSPCT just before the end of the controlling loop.

```
0001          SUBROUTINE SYMBOL(IAY,IAX,FAY,FAX,A,DX,DY,
             *NX,NY,D,DA)
0002          COMMON/DBLOCK/SYMTAB(21,3),JTBLN,RAMTAB(21,3),
             *TEMTAB(21,3),IHEAD,LCOUNT,CHECK
0003            COMMON/DAREA/XMIN,XMAX,YMIN,YMAX
0004            DIMENSION SORT(8,8)
       C
       C
       C
       C        THIS SUBROUTINE WAS WRITTEN BY LT.MARY A. SMART,
       C        AFIT(GCS-80D)FOR A THESIS SPONSORED BY THE
       C        INFORMATION AND PRESENTATION GROUP OF THE AIR
       C        FORCE AVIONICS LAB, WRIGHT-PATTERSON AIR FORCE
       C        BASE,OHIO.  IT IS AN OVERLAY ROUTINE, I.E.,
       C        INFORMATION GENERATED BY THE ROUTINE IS WRITTEN
       C        ON TOP OF INFORMATION GENERATED BY THE MAIN PROGRAM.
       C        THE MAIN USE FOR THIS SUBROUTINE IS IN THE AIRBORNE
       C        ELECTRONIC TERRAIN MAPPING SYSTEM (AETMS) DEVELOPED
       C        BY THE AVIONICS LAB AND WRITTEN BY THE SYSTRAN
       C        CORPORATION.  SYSTRAN DOCUMENTATION GIVES AN OVERVIEW
       C        OF THE AETMS SOFWARE.
       C
       C          THE ENTIRE MODULE CONSISTS OF 15 SUBROUTINES IN THE
       C        FOLLOWING HIERARCHY:
       C
       C        LEVEL 0:   SYMBOL(MAIN)
       C         LEVEL 1:   CONVRT
       C                    CHECK1
       C                    CHECK2
       C                    FLAG
       C          LEVEL 2:   HW
       C                     ELECT
       C                     BRIDA
       C                     BRDB
       C                     TARG
       C                     BLD
       C                     PTR
       C                     LNDM
       C           LEVEL 3:   DRAW
       C                      OFST
       C
       C
       C                  ALGORITHM FOR MAIN
       C
       C        STEP 0:  INITIALIZE ALL VARIABLES WITH ARGUMENTS
       C                 RECEIVED FROM PRSPCT(AETMS).
       C
       C        STEP 1:  CONVERT REAL WORLD LATITUDE AND LONGITUDE
       C                 TO LATITUDE AND LONGITUDE IN PATCH COORDI-
       C                 NATES.
       C
       C        STEP2:  DETERMINE THE HEADING AND CALL THE APPROPRIATE
       C                BOUNDARY CHECKING ROUTINE.
       C
       C        STEP 3:  DETERMINE WHICH SYMBOL BELONGS AT A PARTICULAR
       C                 LATITUDE AND LONGITUDE.
```

```
C
C        STEP 4:  DRAW THE SYMBOL.
C
C        STEP 5:  RETURN TO PRSPCT.
C
C
C
C
C
         JIBLN=21
         RAD=3.14159927/180
         B=1*BAD
         A=A*(1/BAD)
         DA=DA*(1/BAD)
         IHEAD=IFIX((A+DA)+0.5)
         IF(IHEAD .GT. 360)IHEAD=IHEAD-360
         A=A*BAD
         DA=DA*BAD
         A=A+DA
         DH=5.52
C
         PRINT*,'  '
         PRINT*,'AIRCRAFT HEADING AND SPEED'
         TYPE 75,(IHEAD,D)
75       FORMAT(I4,' DEGREES, ',F4.0,' KNOTS')
         PRINT*,'  '
C
C
       POSX=D*(((SIN(A)-(DA*(B/2)*COS(A))))/DX)
       POSY=D*((((DA*(B/2))*SIN(A))+COS(A))/DY)
C
C
       CALL CONVRT(POSX,POSY)
C
       CONS=((DH*SIN(A))/(2*DX))
       CONC=((DH*COS(A))/(2*DY))
C
C
C
       AX=IAX+(FAX)/60
       AY=IAY+(FAY)/60
C
         TYPE 70,(AY,AX)
70       FORMAT(' INITIAL LATITUDE: ',F6.2,' INITIAL LONGITUDE: ',F6.2)
         PRINT*,'  '
C
       QX=IAX+CONS+(FAX+CONS)/60
        IQX=IFIX(QX)
         FQX=(QX-IQX)*60
C
       QY=IAY+CONC+(FAY+CONC)/60
        IQY=IFIX(QY)
         FQY=(QY-IQY)*60
C
       CX=(IAX+(2*CONS))+(FAX+(2*CONS))/60
        ICX=IFIX(CX)
```

```
            FCX=(CX-ICX)*60
      C
           CY=(IAY+(2*CONC))+(FAY+(2*CONC))/60
            ICY=IFIX(CY)
             FCY=(CY-ICY)*60
      C
           BX=(IBX-CONC)+(FAX-CONC)/60
            IBX=IFIX(BX)
             FBX=(BX-IBX)*60
      C
           BY=(IBY+CONS)+(FBY+CONS)/60
            IBY=IFIX(BY)
             FBY=(BY-IBY)*60
      C
           DPX=(IBX+(2*CONC))+(FBX+(2*CONC))/60
            IDPX=IFIX(DPX)
             FDPX=(DPX-IDPX)*60
      C
           DPY=(IBY-(2*CONS))+(FBY-(2*CONS))/60
            IDPY=IFIX(DPY)
             FDPY=(DPY-IDPY)*60
      C
           IWY=(IAY+(DH*COS(A))/(2*DY))-(NY-2)/2
           IWX=(IAX+(DH*SIN(A))/(2*DX))-(NX-2)/2
      C
      C
           IX=IWX
           IY=IWY
      C
           JX=IWX
           JY=IWY+(NY-2)
      C
           KX=IWX+(NX-2)
           KY=IWY+(NY-2)
             PRINT*,'CORNER COORDINATES'
             TYPE*,AX,AY,BX,BY,CX,CY,DPX,DPY
             PRINT*,'    '
      C
                DO 5 M=1,JTBLN
                 DO 5 J=1,3
                  RAMTAB(M,J)=0
                  TEMTAB(M,J)=0
       5         CONTINUE
             SORT(1,1)=AX
             SORT(1,2)=BX
             SORT(1,3)=CX
             SORT(1,4)=DPX
      C
             SORT(2,1)=AY
             SORT(2,2)=BY
             SORT(2,3)=CY
             SORT(2,4)=DPY
      C
             XMAX=0
             XMIN=9999
             DO 17 M=1,4
```

```
          IF(XMIN .LT. SORT(1,M))GOTO 27
          XMIN=SORT(1,M)
          Y1PART=SORT(2,M)
 27         IF(XMAX .GT. SORT(1,M))GOTO 17
          XMAX=SORT(1,M)
          YAPART=SORT(2,M)
 17         CONTINUE
C
          YMAX=0
          YMIN=9999
C
          DO 18 M=1,4
           IF(YMIN .LT. SORT(2,M))GOTO 28
           YMIN=SORT(2,M)
           X1PART=SORT(1,M)
 28         IF(YMAX .GT. SORT(2,M))GOTO 18
           YMAX=SORT(2,M)
           XAPART=SORT(1,M)
 18         CONTINUE
C
          PRINT*,'XMIN,XMAX,YMIN,YMAX'
          TYPE*,XMIN,XMAX,YMIN,YMAX
          PRINT*,'  '
C
        IF((IHEAD .EQ. 0) .OR. (IHEAD .EQ. 90) .OR.
       *(IHEAD .EQ. 180) .OR. (IHEAD .EQ. 270) .OR.
       *(IHEAD .EQ. 360)) CALL CHECK1(XMIN,XMAX,YMIN,YMAX)
C
C
        IF((IHEAD .NE. 0) .AND. (IHEAD .NE. 90) .AND.
       *(IHEAD .NE. 180) .AND. (IHEAD .NE. 270) .AND.
       *(IHEAD .NE. 360))CALL CHECK2(AX,AY,BX,BY,CX,CY,DPX,DPY,QX,
       *QY,XMIN,XMAX,YMIN,YMAX)
C
        IF(LCOUNT .EQ. 0)GOTO 10
C
          PRINT*,' THE RAMTEK TABLE'
         DO 777 M=1,LCOUNT
          TYPE 778,(RAMTAB(M,J),J=1,3)
 778      FORMAT(3(2X,F9.4))
 777     CONTINUE
        CALL FLAG(LCOUNT,RAMTAB)
C
 10     RETURN
        END
```

```
C
C
C
      SUBROUTINE CHECK1(XMIN,XMAX,YMIN,YMAX)
      COMMON/DBLOCK/SYMTAB(21,3),JTBLN,RAMTAB(21,3),
     *TEMTAB(21,3),IHEAD,LCOUNT,CHECK
C
C
C                 ALGORITHM(CHECK1)
C
C        STEP 0:  INITIALIZE VARIABLES WITH ARGUMENTS
C                 RECEIVED FROM MAIN.
C
C        STEP 1:  DO FOR ALL VALUES:
C                  COMPARE TABLE VALUES AGAINST MAXIMUM
C                  AND MINIMUM VALUES.  IF VALUE IS WITHIN
C                  THESE VALUES PLACE THE VALUE IN THE RAMTEK
C                  TABLE.
C
C        STEP 2:  COUNT THESE VALUES.
C
C        STEP 3:  RETURN TO MAIN.
C
C
C
            LCOUNT=0
            K=0
            J=1
            DO 10 I=1,JTBLN
            XVAL=SYMTAB(I,1)
            YVAL=SYMTAB(I,2)
            ZVAL=SYMTAB(I,3)
C
             IF((YVAL .GT. YMAX) .OR. (YVAL .LT. YMIN)
     *.OR. (XVAL .GT. XMAX) .OR. (XVAL .LT. XMIN))
     *GOTO 10
             RAMTAB(J,1)=XVAL
             RAMTAB(J,2)=YVAL
             RAMTAB(J,3)=ZVAL
             K=K+1
             J=J+1
             LCOUNT=LCOUNT+1
   10       CONTINUE
      IF(K .EQ. 0)LCOUNT=0
      CHECK=1.0
      RETURN
      END
```

```
C
C
C
      SUBROUTINE CHECK2(AX,AY,BX,BY,CX,CY,DPX,DPY,QX,QY,XMIN,
     *XMAX,YMIN,YMAX)
      COMMON/DBLOCK/SYMTAB(21,3),JTHEN,RAMTAB(21,3),
     *TEMTAB(21,3),IHEAD,LCOUNT,CHECK
C
C
C                  ALGORITHM(CHECK2)
C
C        STEP 0:  INITIALIZE VARIABLES WITH ARGUMENTS
C                 RECEIVED FROM MAIN.
C
C        STEP 1:  DO FOR ALL VALUES:
C                  DETERMINE IF VALUE IS IN WINDOW.  IF VALUE IS
C                  WITHIN THESE BOUNDS, PLACE THE VALUE IN A TEMPORARY
C                  TABLE.
C        STEP 2:  COUNT THESE VALUES.
C
C        STEP 3:  DO FOR ALL VALUES IN TEMP:
C                 CHECK THESE VALUES AGAINST THE ROTATED BOUNDS.
C                 IF THE VALUE IS WITHIN THE BOUNDS, PLACE THE
C                 VALUE IN THE RAMTEK TABLE.
C
C        STEP 4:  COUNT THESE VALUES.
C
C        STEP 5:  RETURN TO MAIN.
C
C
        IF(AY .NE. YMIN)GOTO 5
      SAB=(BY-AY)/(BX-AX)
      SBC=(CY-BY)/(CX-BX)
      SDC=(DPY-CY)/(DPX-CX)
      SAD=(AY-DPY)/(AX-DPX)
C
C
        SAXY=SAB
        SBXY=SBC
        SCXY=SDC
        SDXY=SAD
C
 5       IF(DPY .NE. YMIN)GOTO 10
          SAB=(DPY-AY)/(DPX-AX)
          SBC=(AY-BY)/(AX-BX)
          SDC=(BY-CY)/(BX-CX)
          SAD=(CY-DPY)/(CX-DPX)
C
        SAXY=SBC
        SBXY=SDC
        SCXY=SAD
        SDXY=SAB
C
 10        IF(CY .NE. YMIN)GOTO 15
            SAB=(CY-DPY)/(CX-DPX)
            SBC=(DPY-AY)/(DPX-AX)
```

```
            SDC=(AY-BY)/(AX-BX)
            SAD=(BY-CY)/(BX-CX)
C
         SAXY=SDC
         SBXY=SAD
         SCXY=SAB
         SDXY=SBC
C
   15      IF(BY .NE. YMIN)GOTO 35
            SAB=(BY-CY)/(BX-CX)
            SBC=(CY-DPY)/(CX-DPX)
            SDC=(DPY-AY)/(DPX-AX)
            SAD=(BY-AY)/(BX-AX)
C
         SAXY=SAD
         SBXY=SAB
         SCXY=SBC
         SDXY=SDC
C
C
C
      PM1X=(XMIN+XAPART)/2
      PM1Y=(YIPART+YMAX)/2
C
      PM2X=(XAPART+XMAX)/2
      PM2Y=(YMAX+YAPART)/2
C
      PM3X=(XMAX+XIPART)/2
      PM3Y=(YAPART+YMIN)/2
C
      PM4X=(XIPART+XMIN)/2
      PM4Y=(YMIN+YIPART)/2
C
      C1X=(XAPART+QX)/2
      C1Y=(YMAX+QY)/2
C
      C2X=(XMAX+QX)/2
      C2Y=(YAPART+QY)/2
C
      C3X=(XIPART+QX)/2
      C3Y=(YMIN+QY)/2
C
      C4X=(XMIN+QX)/2
      C4Y=(YIPART+QY)/2
C
C
C
            LCOUNT=0
            CHECK=0.0
             K=0
             J=1
            DO 20 I=1,JTBLN
             XVAL=SYMTAB(I,1)
             YVAL=SYMTAB(I,2)
             ZVAL=SYMTAB(I,3)
C
```

```
0063            IF((YVAL .GT. YMAX) .OR. (YVAL .LT. YMIN)
           *   .OR. (XVAL .GT. XMAX) .OR. (XVAL .LT. XMIN))
           *    GOTO 20
        C
0064            TEMTAB(J,1)=XVAL
0065            TEMTAB(J,2)=YVAL
0066            TEMTAB(J,3)=ZVAL
0067            K=K+1
0068            J=J+1
0069     20    CONTINUE
        C
        C
        C
0070          IF(K .EQ. 0)GOTO 35
0071          J=1
0072            DO 30 I=1,K
0073               X=TEMTAB(I,1)
0074               Y=TEMTAB(I,2)
0075               Z=TEMTAB(I,3)
        C
0076          IF(((X .EQ. AX) .OR. (X .EQ. BX) .OR. (X .EQ. CX)
           *.OR. (X .EQ. DPX)) .AND. ((Y .EQ. AY) .OR.
           *(Y .EQ. BY) .OR. (Y .EQ. CY) .OR. (Y .EQ. DPY)))
           *GOTO 25
        C
0077            IF((PM1X .LE. X) .AND. (X .LE. C1X) .AND.
           *   (Y .GE. C1Y)) SAXY=(Y-YIPART)/(X-XMIN)
        C
0078            IF((C1X .LE. X) .AND. (X .LE. PM2X) .AND.
           *    (Y .GE. C1Y)) SBXY=(Y-YMAX)/(X-XAPART)
        C
0079            IF((C2Y .LE.Y) .AND. (Y .LE. PM2Y) .AND.
           *    (X .GE. C2X)) SBXY=(Y-YMAX)/(X-XAPART)
        C
0080            IF((PM3Y .LE. Y) .AND. (Y .LE. C2Y) .AND.
           *   (X .GE. C2X)) SCXY=(Y-YAPART)/(X-XMAX)
        C
0081            IF((C3X .LE. X) .AND. (X .LE. PM3X) .AND.
           *    (Y .LE. C3Y)) SCXY=(Y-YAPART)/(X-XMAX)
        C
0082            IF((PM4X .LE. X) .AND. (X .LE. C3X) .AND.
           *    (Y .LE. C3Y)) SDXY=(Y-YMIN)/(X-XIPART)
        C
0083            IF((PM4Y .LE. Y) .AND. (Y .LE. C4Y) .AND.
           *    (X .LE. C4X)) SDXY=(Y-YMIN)/(X-XIPART)
        C
0084            IF((C4Y .LE. Y) .AND. (Y .LE. PM1Y) .AND.
           *    (X .LE. C4X)) SAXY=(Y-YIPART)/(X-XMIN)
        C
        C
        C
0085            IF((SAXY .LT. SBC) .OR. (SBXY .GT. SDC) .OR.
           *   (SCXY .GT. SAD) .OR. (SDXY .LT. SAB)) GOTO 30
        C
0086     25     RAMTAB(J,1)=X
0087            RAMTAB(J,2)=Y
```

```
0088              PAMTAB(J,3)=Z
0089               LCOUNT=LCOUNT+1
0090               J=J+1
0091    30        CONTINUE
       C
0092    35     RETURN
0093           END
```

```
      C
      C
0001        BLOCK DATA
0002        COMMON/PTABLE/POSTAB(21,3)
      C
      C
      C       THIS DATA IS VALUES OF REAL WORLD LATITUDE AND
      C       LONGITUDE.
      C
      C
0003        DATA POSTAB/15.5,25.5,35.5,45.5,55.5,65.5,75.5,
           *85.50,86.79,87.00,83.12,83.59,84.10,83.9,83.51,84.26,
           *85.09,85.22,86.13,86.61,85.0,10.75,20.75,30.75,40.75,50.75,
           *60.76,70.75,80.75,35.22,35.39,35.59,35.45,35.16,34.21,34.0,
           *34.26,34.59,34.16,34.76,34.01,35.0,20.0,30.0,40.0,50.0,
           *60.0,70.0,80.0,90.0,90.0,30.0,20.0,20.0,20.0,20.0,20.0,
           *20.0,20.0,20.0,20.0,20.0,20.0/
0004        END
```

```
C
C
C
      SUBROUTINE CONVRT(POSX,POSY)
      COMMON/PTABLE/POSTAB(21,3)
      COMMON/DBLOCK/SYMTAB(21,3),JTBLN,RADTAB(21,3),
     *TEMTAB(21,3),IHEAD,LCOUNT,CHECK
C
C
C                   ALGORITHM(CONVRT)
C
C        STEP 0:  INITIALIZE VARIABLES WITH ARGUMENTS RECEIVED
C                 FROM MAIN.
C
C        STEP 1:  RETRIEVE DATA FROM BLOCK DATA.
C
C        STEP 2:  DO FOR ALL VALUES:
C                  ADD THE OFFSET VALUE TO CONVERT VALUES TO
C                  VALUES IN PATCH COORDINATES.
C
C        STEP 3:  RETURN TO MAIN.
C
       J=1
       DO 30 I=1,JTBLN
           X=POSTAB(I,1)
           Y=POSTAB(I,2)
           Z=POSTAB(1,3)
C
             IFX=IFIX(X)
              FX=(X-IFX)*60
C
             IFY=IFIX(Y)
              FY=(Y-IFY)*60
C
              IFX=IFX+POSX
              FX=FX+POSX
C
              IFY=IFY+POSY
              FY=FY+POSY
C
              X=IFX+(FX/60)
C
              Y=IFY+(FY/60)
C
           SYMTAB(J,1)=X
           SYMTAB(J,2)=Y
           SYMTAB(J,3)=Z
           J=J+1
 30     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FLAG(LCOUNT,RAMTAB)
      DIMENSION RAMTAB(21,3)
C
C
C
C                    ALGORITHM(FLAG)
C
C       STEP 0:  INITIALIZE VARIABLES WITH VALUES RECEIVED
C                FROM MAIN.
C
C       STEP 1:  DO FOR ALL VALUES IN THE RAMTEK TABLE:
C                 CHECK THE FLAG VALUE (RAMTAB(L,3)) TO FIND
C                 WHICH SYMBOL IS TO BE DRAWN.
C
C       STEP 2:  RETURN TO MAIN AFTER ALL SYMBOLS ARE DRAWN.
C
C
C
      L=1
    4 IF(L .GT. LCOUNT)GOTO 999
      J=L
        IFLAG=0
        IF(L .EQ. 1)IFLAG=1
C
       IF(RAMTAB(L,3) .NE. 20)GOTO 7
        CALL HW(J,RAMTAB,IFLAG)
        IFLAG=0
        PRINT*,'HIGHWAY HAS BEEN  DRAWN'
        GOTO 30
C
    7  IF(RAMTAB(L,3) .NE. 30)GOTO 8
        CALL ELECT(J,RAMTAB,IFLAG)
        IFLAG=0
        PRINT*,'ELECTRICAL LINES HAVE BEEN DRAWN'
        GOTO 30
C
    8  IF(RAMTAB(L,3) .NE. 40)GOTO 9
        CALL BRIDA(J,RAMTAB,IFLAG)
        IFLAG=0
        PRINT*,'BRIDGE A HAS BEEN DRAWN'
        GOTO 30
C
    9  IF(RAMTAB(L,3) .NE. 50)GOTO 10
        CALL BRDB(J,RAMTAB,IFLAG)
        IFLAG=0
        PRINT*,'BRIDGE B HAS BEEN DRAWN'
        GOTO 30
C
   10  IF(RAMTAB(L,3) .NE. 60)GOTO 12
        CALL PTR(J,RAMTAB,IFLAG)
        IFLAG=0
        PRINT*,'THE POINTER HAS BEEN DRAWN'
        GOTO 30
C
   12  IF(RAMTAB(L,3) .NE. 70)GOTO 13
        CALL LNDST(J,RAMTAB,IFLAG)
```

```
      IFLAG=0
      PRINT*,'THE LANDING STRIP HAS BEEN DRAWN'
      GOTO 30
C
13    IF(RAMTAB(L,3) .NE. 80)GOTO 14
      CALL TARG(J,RAMTAB,IFLAG)
      IFLAG=0
      PRINT*,'THE TARGET SYMBOL HAS BEEN DRAWN'
      GOTO 30
C
14    IF(RAMTAB(L,3) .NE. 90)GOTO 15
       CALL BLD(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
15     PRINT*,'ERROR IN FLAG'
C
30    L=L+1
      GOTO 4
999   RETURN
      END
```

```
              C
              C
0001                SUBROUTINE DRAW(MAT,IFLAG)
              C
              C        THIS SUBROUTINE WAS ADAPTED FROM A PROGRAM
              C        WRITTEN BY DALE RANGELER OF SRL,INC.
              C        PREVIOUS PROGRAM NAME:  FINAL,V3.0,
              C         9 SEPTEMBER 1980.
              C
0002                IMPLICIT INTEGER(A-Z)
0003                DIMENSION RAMDAT(10),RAMPRM(6),IOSTAT(2),MAT(5,5)
0004                BYTE IOBYTE(2)
0005                EQUIVALENCE (IOBYTE(1),IOSTAT(1))
0006                WLB="400
0007                RSET="2400
0008                INOP="3400
0009                MSK=1
0010                FGD=2
0011                COP="100000
0012                WV="7000
0013                DF=1
0014                UF=2
              C
0015                CALL ASNLUN(1,'XR',0)
0016                RAMDAT(1)=INOP
0017                IF(IFLAG .EQ. 1)RAMDAT(1)=RSET
0018                RAMDAT(2)=(WV.OR.UF.OR.DF)
0019                RAMDAT(3)=(COP.OR.FGD.OR.MSK)
0020                RAMDAT(4)="3400
0021                RAMDAT(5)="2000
0022                RAMDAT(6)=MAT(1,1)
0023                RAMDAT(7)=MAT(1,2)
0024                RAMDAT(8)=4
0025                RAMDAT(9)=MAT(2,1)
0026                RAMDAT(10)=MAT(2,2)
              C
0027                CALL GETADR(RAMPRM,RAMDAT)
0028                RAMPRM(2)=20
0029                RAMPRM(3)=0
              C
0030                CALL WTQIO(WLB,1,2,3,IOSTAT,RAMPRM,IER)
              C
0031                TYPE*,'DIRECTIVE STATUS=',IER
0032                TYPE*,'IOSTAT =',IOBYTE(1)
              C
0033                RETURN
0034                END
```

```
C
C
C        THE FOLLOWING ALGORITHM IS  THE SAME FOR ALL SYMBOL
C        DRAWING SUBROUTINES.
C
C                  ALGORITHM
C
C        STEP 0:  INITIALIZE VARIABLES WITH ARGUMENTS
C                 FROM FLAG.
C
C        STEP 1:  SCALE THE STARTING POINT TO VIDEO SCREEN SIZE.
C
C        STEP 2:  DRAW THE PICTURE.
C
C        STEP 3:  RETURN TO FLAG.
C
C
      SUBROUTINE HW(J,RAMTAB,IFLAG)
      IMPLICIT INTEGER(H)
      DIMENSION RAMTAB(21,3),HWM(5,5),RAMTBL(21,3)
C
      ICOUNT=3
        L=J
          CALL OFST(L,RAMTAB,RAMTBL)
C
 10       HWM(1,1)=IFIX(RAMTBL(J,1)+0.5)
          HWM(1,2)=IFIX(RAMTBL(J,2)+0.5)
          HWM(2,1)=HWM(1,1)+03
          HWM(2,2)=HWM(1,2)
C
        CALL DRAW(HWM,IFLAG)
        IFLAG=0
        ICOUNT=ICOUNT-1
      IF(ICOUNT .EQ. 0)GOTO 250
      RAMTBL(J,2)=RAMTBL(J,2)+14
      GOTO 10
 250  RETURN
      END
```

```
      SUBROUTINE ELECT(J,RAMTAB,IFLAG)
      IMPLICIT INTEGER(E)
      DIMENSION RAMTAB(21,3),ELEC(5,5),RAMTBL(21,3)
       ICOUNT=0
       CALL OFST(J,RAMTAB,RAMTBL)
4      ELEC(1,1)=IFIX(RAMTBL(J,1)+0.5)
       ELEC(1,2)=RAMTBL(J,2)
       ELEC(2,1)=ELEC(1,1)+23
       ELEC(2,2)=ELEC(1,2)
C
       RAMTBL(J,1)=ELEC(2,1)+36
        CALL DRAW(ELEC,IFLAG)
        IFLAG=0
        ICOUNT=ICOUNT+1
       IF(ICOUNT .GT. 5)GOTO 999
C
       GOTO 4
999    RETURN
       END
```

```
      C
      C
      C
0001        SUBROUTINE BRDA(J,RAMTAB,IFLAG)
0002        IMPLICIT INTEGER(B)
0003        DIMENSION RAMTAB(21,3),BRDA(5,5),HOLD(5),RAMTBL(21,3)
      C
0004         CALL OFST(J,RAMTAB,RAMTBL)
      C
0005         BRDA(1,1)=IFIX(RAMTBL(J,1)+0.5)
0006         BRDA(1,2)=IFIX(RAMTBL(J,2)+0.5)
0007         BRDA(2,1)=BRDA(1,1)+23
0008         HOLD(1)=BRDA(2,1)
0009         BRDA(2,2)=BRDA(1,2)-23
0010         HOLD(2)=BRDA(2,2)
      C
0011         CALL DRAW(BRDA,IFLAG)
0012         IFLAG=0
      C
0013         BRDA(1,1)=HOLD(1)
0014         BRDA(1,2)=HOLD(2)
0015         BRDA(2,1)=BRDA(1,1)+35
0016         HOLD(1)=BRDA(2,1)
0017         BRDA(2,2)=BRDA(1,2)
0018         HOLD(2)=BRDA(2,2)
      C
0019          CALL DRAW(BRDA,IFLAG)
      C
0020          BRDA(1,1)=HOLD(1)
0021          BRDA(1,2)=HOLD(2)
0022          BRDA(2,1)=BRDA(1,1)+23
0023          BRDA(2,2)=BRDA(1,2)+23
      C
0024          CALL DRAW(BRDA,IFLAG)
0025          RETURN
0026          END
```

```
C
C
C
      SUBROUTINE BRDR(J,RAMTAB,IFLAG)
      IMPLICIT INTEGER(B)
      DIMENSION RAMTAB(21,3),BRD(5,5),HOLD(5),RAMTBL(21,3)
C
       CALL OFST(J,RAMTAB,RAMTBL)
C
       BRD(1,1)=IFIX(RAMTBL(J,1)+0.5)
       BRD(1,2)=IFIX(RAMTBL(J,2)+0.5)
       BRD(2,1)=BRD(1,1)+23
       HOLD(1)=BRD(2,1)
       BRD(2,2)=BRD(1,2)+23
       HOLD(2)=BRD(2,2)
C
       CALL DRAW(BRD,IFLAG)
C
       IFLAG=0
C
      BRD(1,1)=HOLD(1)
      BRD(1,2)=HOLD(2)
      BRD(2,1)=BRD(1,1)+35
      HOLD(1)=BRD(2,1)
      BRD(2,2)=BRD(1,2)
      HOLD(2)=BRD(2,2)
C
       CALL DRAW(BRD,IFLAG)
C
       BRD(1,1)=HOLD(1)
       BRD(1,2)=HOLD(2)
       BRD(2,1)=BRD(1,1)+23
       BRD(2,2)=BRD(1,2)-23
C
       CALL DRAW(BRD,IFLAG)
C
       BRD(1,1)=IFIX(RAMTBL(J,1)+0.5)
       BRD(1,2)=IFIX(RAMTBL(J,2)+0.5)+69
       BRD(2,1)=BRD(1,1)+23
       HOLD(1)=BRD(2,1)
       BRD(2,2)=BRD(1,2)-23
       HOLD(2)=BRD(2,2)
C
       CALL DRAW(BRD,IFLAG)
C
       BRD(1,1)=HOLD(1)
       BRD(1,2)=HOLD(2)
       BRD(2,1)=BRD(1,1)+35
       HOLD(1)=BRD(2,1)
       BRD(2,2)=BRD(1,2)
       HOLD(2)=BRD(2,2)
C
       CALL DRAW(BRD,IFLAG)
C
       BRD(1,1)=HOLD(1)
       BRD(1,2)=HOLD(2)
```

```
      BRD(2,1)=BRD(1,1)+23
      BRD(2,2)=BRD(1,2)+23
C
      CALL DRAW(BRD,IFLAG)
      RETURN
      END
```

```
C
C
C
      SUBROUTINE PTR(J,RAMTAB,IFLAG)
      IMPLICIT INTEGER(P)
      DIMENSION RAMTAB(21,3),PTRM(5,5),HOLD(5),RAMTBL(21,3)
C
      CALL OFST(J,RAMTAB,RAMTBL)
C
       PTRM(1,1)=IFIX(RAMTBL(J,1)+0.5)
       PTRM(1,2)=IFIX(RAMTBL(J,2)+0.5)
       PTRM(2,1)=PTRM(1,1)+150
       HOLD(1)=PTRM(2,1)
       PTRM(2,2)=PTRM(1,2)
       HOLD(2)=PTRM(2,2)
C
       CALL DRAW(PTRM,IFLAG)
C
       IFLAG=0
C
       PTRM(1,1)=HOLD(1)
       PTRM(1,2)=HOLD(2)
       PTRM(2,1)=PTRM(1,1)-43
       HOLD(1)=PTRM(2,1)
       PTRM(2,2)=PTRM(1,2)-43
       HOLD(2)=PTRM(2,2)
C
        CALL DRAW(PTRM,IFLAG)
C
         PTRM(1,1)=HOLD(1)
         PTRM(1,2)=HOLD(2)+86
         PTRM(2,1)=PTRM(1,1)+43
         PTRM(2,2)=PTRM(1,2)-43
C
       CALL DRAW(PTRM,IFLAG)
C
      RETURN
      END
```

```
C
C
C
      SUBROUTINE LNDST(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),LNDM(5,5),HOLD(5),RAMTBL(21,3)
C
      CALL OFST(J,RAMTAB,RAMTBL)
C
      LNDM(1,1)=IFIX(RAMTBL(J,1)+0.5)
      LNDM(1,2)=IFIX(RAMTBL(J,2)+0.5)
      LNDM(2,1)=LNDM(1,1)+75
      HOLD(1)=LNDM(2,1)
      LNDM(2,2)=LNDM(1,2)
      HOLD(2)=LNDM(2,2)
C
       CALL DRAW(LNDM,IFLAG)
C
      IFLAG=0
       LNDM(1,1)=HOLD(1)-50
       LNDM(1,2)=HOLD(2)+25
       LNDM(2,1)=LNDM(1,1)
       HOLD(1)=LNDM(2,1)
       LNDM(2,2)=LNDM(1,2)-75
       HOLD(2)=LNDM(2,2)
C
       CALL DRAW(LNDM,IFLAG)
C
      LNDM(1,1)=HOLD(1)
      LNDM(1,2)=HOLD(2)
      LNDM(2,1)=LNDM(1,1)+25
      LNDM(2,2)=LNDM(1,2)+75
C
       CALL DRAW(LNDM,IFLAG)
      RETURN
      END
```

```
C
C
C
      SUBROUTINE TARG(J,RAMTAB,IFLAG)
      IMPLICIT INTEGER(A-C)
      DIMENSION RAMTAB(21,3),ATAR(5,5),CONST(5),RAMTBL(21,3)
C
      CALL OFST(J,RAMTAB,RAMTBL)
C
      CONST(1)=IFIX(RAMTBL(J,1)+0.5)
      CONST(2)=IFIX(RAMTBL(J,2)+0.5)
C
      ATAR(1,1)=CONST(1)+13
      ATAR(1,2)=CONST(2)
      ATAR(2,1)=ATAR(1,1)+38
      ATAR(2,2)=ATAR(1,2)
C
      CALL DRAW(ATAR,IFLAG)
C
      IFLAG=0
C
     ATAR(1,1)=CONST(1)
     ATAR(1,2)=CONST(2)+13
     ATAR(2,1)=ATAR(1,1)
     ATAR(2,2)=ATAR(1,2)+38
C
      CALL DRAW(ATAR,IFLAG)
C
      ATAR(1,1)=CONST(1)
      ATAR(1,2)=CONST(2)-13
      ATAR(2,1)=ATAR(1,1)
      ATAR(2,2)=ATAR(1,2)-38
C
      CALL DRAW(ATAR,IFLAG)
C
      ATAR(1,1)=CONST(1)-13
      ATAR(1,2)=CONST(2)
      ATAR(2,1)=ATAR(1,1)-38
      ATAR(2,2)=ATAR(1,2)
C
      CALL DRAW(ATAR,IFLAG)
C
      RETURN
      END
```

```
C
C
C
      SUBROUTINE BLD(J,RAMTAB,IFLAG)
      IMPLICIT INTEGER(B,T)
      DIMENSION RAMTAB(21,3),BLDM(5,5),TEMP(5),HOLD(5),RAMTBL(21,3)
C
      CALL OFST(J,RAMTAB,RAMTBL)
C
      ICOUNT=0
5      BLDM(1,1)=IFIX(RAMTBL(J,1)+0.5)
       BLDM(1,2)=IFIX(RAMTBL(J,2)+0.5)
       BLDM(2,1)=BLDM(1,1)
       HOLD(1)=BLDM(2,1)
       BLDM(2,2)=BLDM(1,2)-75
       HOLD(2)=BLDM(2,2)
C
       CALL DRAW(BLDM,IFLAG)
C
      IFLAG=0
      ICOUNT=ICOUNT+1
C
       BLDM(1,1)=HOLD(1)
       BLDM(1,2)=HOLD(2)
       BLDM(2,1)=BLDM(1,1)+23
       HOLD(1)=BLDM(2,1)
       BLDM(2,2)=BLDM(1,2)
       HOLD(2)=BLDM(2,2)
C
       CALL DRAW(BLDM,IFLAG)
C
       ICOUNT=ICOUNT+1
       IF(ICOUNT .GE. 5) GOTO 6
       BLDM(1,1)=HOLD(1)
       BLDM(1,2)=HOLD(2)
       BLDM(2,1)=BLDM(1,1)
       BLDM(2,2)=BLDM(1,2)+75
C
       CALL DRAW(BLDM,IFLAG)
C
       ICOUNT=ICOUNT+1
C
       TEMP(1)=IFIX(RAMTBL(J,1)+0.5)
       TEMP(2)=IFIX(RAMTBL(J,2)+0.5)
       RAMTBL(J,1)=TEMP(1)+23
       RAMTBL(J,2)=TEMP(2)
       GOTO 5
C
6       BLDM(1,1)=IFIX(RAMTBL(J,1)+0.5)+23
        BLDM(1,2)=IFIX(RAMTBL(J,2)+0.5)-75
        BLDM(2,1)=BLDM(1,1)
        HOLD(1)=BLDM(2,1)
        BLDM(2,2)=BLDM(1,2)+45
        HOLD(2)=BLDM(2,2)
C
       CALL DRAW(BLDM,IFLAG)
```

```
C
      BLDM(1,1)=HOLD(1)
      BLDM(1,2)=HOLD(2)
      BLDM(2,1)=BLDM(1,1)+35
      HOLD(1)=BLDM(2,1)
      BLDM(2,2)=BLDM(1,2)
      HOLD(2)=BLDM(2,2)
C
C
      CALL DRAW(BLDM,IFLAG)
C
      BLDM(1,1)=HOLD(1)
      BLDM(1,2)=HOLD(2)
      BLDM(2,1)=BLDM(1,1)
      HOLD(1)=BLDM(2,1)
      BLDM(2,2)=BLDM(1,2)+30
      HOLD(2)=BLDM(2,2)
C
      CALL DRAW(BLDM,IFLAG)
C
      BLDM(1,1)=TEMP(1)
      BLDM(1,2)=TEMP(2)
      BLDM(2,1)=HOLD(1)
      BLDM(2,2)=HOLD(2)
C
      CALL DRAW(BLDM,IFLAG)
C
      RETURN
      END
```

```
C
C
C
      SUBROUTINE OFST(L,RAMTAB,RAMTBL)
      COMMON/DARLA/XMIN,XMAX,YMIN,YMAX
      DIMENSION RAMTAB(21,3),RAMTBL(21,3)
C
C
C                ALGORITHM(OFST)
C
C     STEP 0:  INITIALIZE VARIABLES WITH ARGUMENTS FROM
C              THE SYMBOL DRAWING ROUTINE.
C
C     STEP 1:  ADD OFFSET VALUE.
C
C     STEP 2:  RETURN TO THE APPROPRIATE ROUTINE.
C
C
      RAMTBL(L,1)=480.*((RAMTAB(L,1)-XMIN)/(XMAX-XMIN))
      RAMTBL(L,2)=640.*((RAMTAB(L,2)-YMIN)/(YMAX-YMIN))
C
      RETURN
      END
```

```
      SUBROUTINE FLAG(LCOUNT,RAMTAB)
      DIMENSION RAMTAB(21,3)
C
C
C     THIS ROUTINE HAS THE SAME ALGORITHM AS THE
C     FIRST FLAG.
C
      L=1
4     IF(L .GT. LCOUNT)GOTO 999
      J=L
      IFLAG=0
      IF(L .EQ. 1)IFLAG=1
C
      IF(RAMTAB(L,3) .NE. 95)GOTO 16
       CALL FFTBRD(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
16    IF(RAMTAB(L,3) .NE. 105)GOTO 17
       CALL FFTELE(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
17    IF(RAMTAB(L,3) .NE. 115)GOTO 18
       CALL FFTPTR(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
18    IF(RAMTAB(L,3) .NE. 125)GOTO 19
       CALL FFTARG(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
19    PRINT*,'ERROR IN FLAG'
30    L=L+1
C
C
      GOTO 4
999   RETURN
      END
```

```
      SUBROUTINE FLAG(LCOUNT,RAMTAB)
      DIMENSION RAMTAB(21,3)
C
C     DUE TO ADDRESS OVERFLOW IN THE SYSTEM, IT WAS
C     NECESSARY TO BREAK THE FFT GENERATING
C     PROGRAM INTO TWO SEPERATE PROGRAMS.  PROGRAM 1
C     IS EXACTLY THE SAME AS PROGRAM 2 EXCEPT THAT
C     THEY GENERATE FOUR DIFFERENT SYMBOLS AS
C     FOLLOWS;  PROGRAM 1 GENERATES
C     FFTHW,FFTBRH,FFTLND,AND FFTBLD;  PROGRAM 2
C     GENERATES FFTPTR,FFTTAR,FFTBRD, AND FFTELE.
C
C
C               ALGORITHM(FLAG)
C
C     STEP 0:  INITIALIZE VARIABLES WITH ARGUMENTS
C              RECEIVED FROM MAIN.
C
C     STEP 1:  DO FOR ALL VALUES;
C               CHECK THE FLAG (RAMTAB(L,3)) AND CALL
C               THE APPROPRIATE SYMBOL DRAWING ROUTINE.
C
C     STEP 2:  RETURN TO MAIN AFTER ALL SYMBOLS HAVE BEEN
C              DRAWN.
C
C
      L=1
    4 IF(L .GT. LCOUNT)GOTO 999
      J=L
      IFLAG=0
       IF(L .EQ. 1)IFLAG=1
C
      IF(RAMTAB(L,3) .NE. 95)GOTO 16
       CALL FFTBRB(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
   16 IF(RAMTAB(L,3) .NE. 105)GOTO 17
       CALL FFTHW(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
   17 IF(RAMTAB(L,3) .NE. 115)GOTO 18
       CALL FFTLND(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
   18 IF(RAMTAB(L,3) .NE. 125)GOTO 19
       CALL FFTBLD(J,RAMTAB,IFLAG)
       IFLAG=0
       GOTO 30
C
   19 PRINT*,'ERROR IN FLAG'
C
   30 L=L+1
C
```

```
      GOTO 4
C
  999 RETURN
      END
```

```
      SUBROUTINE SCALE(MAT,IMAT)
      DIMENSION MAT(128,2),IMAT(128,2)
      XMIN=MAT(1,1)
      XMAX=XMIN
      YMIN=MAT(1,2)
      YMAX=YMIN
      DO 10 I=1,128
       IF(MAT(I,1) .LT. XMIN)XMIN=MAT(I,1)
       IF(MAT(I,1) .GT. XMAX)XMAX=MAT(I,1)
       IF(MAT(I,2) .LT. YMIN)YMIN=MAT(I,2)
       IF(MAT(I,2) .GT. YMAX)YMAX=MAT(I,2)
   10 CONTINUE
        DO 20 I=1,128
         IMAT(I,1)=IFIX(((MAT(I,1)-XMIN)/(XMAX-XMIN))*40+0.5)
         IMAT(I,2)=IFIX(((MAT(I,2)-YMIN)/(YMAX-YMIN))*64+0.5)
   20   CONTINUE
       RETURN
       END
```

```
      SUBROUTINE FFTHW(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),NN(2),WORK(56),RMAT(128,2)
     DIMENSION IMAT(128,2),X(128),Y(128),IHWM(5,5),RAMTBL(21,3)
     COMPLEX DATA(128,2)
     DATA NN/128,2/
      IPTS=128
      ILEG=128/3
      RNTVL=0.9/(IPTS-1)
     L=J
     CALL OFST(L,RAMTAB,RAMTBL)
       Y(1)=RAMTBL(J,2)
       X(1)=RAMTBL(J,2)
         DO 10 I=2,ILEG
          X(I)=X(I-1)+RNTVL
          Y(I)=Y(I-1)
   10    CONTINUE
       KK=IPTS-ILEG-1
       K=ILEG+1
        X(K)=X(1)
      Y(K)=Y(ILEG)+RNTVL
        DO 20 I=K,KK
         X(I)=X(I-1)+RNTVL
         Y(I)=Y(I-1)
   20   CONTINUE
      K=KK+1
      Y(K)=Y(KK)+RNTVL
      X(K)=X(1)
         DO 30 I=K,IPTS
          X(I)=X(I-1)+RNTVL
        . Y(I)=Y(I-1)
   30    CONTINUE
        DO 40 J=1,128
         DATA(J,1)=CMPLX(X(J),0.0)
         DATA(J,2)=CMPLX(Y(J),0.0)
   40   CONTINUE
         CALL FOURT(DATA,NN,2,1,0,WORK)
         CALL ZERO(DATA)
         CALL FOURT(DATA,NN,2,-1,1,WORK)
        DO 50 I=1,128
         RMAT(I,1)=REAL(DATA(I,1))
         RMAT(I,2)=REAL(DATA(I,2))
   50   CONTINUE
         CALL SCALE(RMAT,IMAT)
        DO 60 I=2,128
         IHWM(1,1)=IMAT(I-1,1)
         IHWM(1,2)=IMAT(I-1,2)
         IHWM(2,1)=IMAT(I,1)
         IHWM(2,2)=IMAT(I,2)
         CALL DRAW(IHWM,IFLAG)
         IFLAG=0
   60   CONTINUE
         RETURN
         END
```

```
      SUBROUTINE FFILND(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
      DIMENSION RMAT(128,2),IMAT(128,2),ILND(5,5),NN(2),WORK(50)
C
      COMPLEX DATA(128,2)
      DATA NN/128,2/
      L=J
      IPTS=128
      ILEG=128/3
      RNTVL=.9/(IPTS-1)
C
      CALL UFST(L,RAMTAB,RAMTBL)
      X(1)=RAMTBL(J,1)
      Y(1)=RAMTBL(J,2)
      DO 30 I=2,ILEG
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
30    CONTINUE
C
      K=ILEG+1
      KK=IPTS-ILEG-1
      Y(K)=Y(1)-RNTVL
      X(K)=X(1)+RNTVL
C
      K=K+1
      DO 40 I=K,KK
       X(I)=X(I-1)-(0.5*RNTVL)
       Y(I)=Y(I-1)+RNTVL
40    CONTINUE
C
      K=KK+1
      DO 50 I=K,IPTS
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)-RNTVL
50      CONTINUE
C
      DO 70 I=1,128
       DATA(I,1)=CMPLX(X(I),0.0)
       DATA(I,2)=CMPLX(Y(I),0.0)
70    CONTINUE
      CALL FOURT(DATA,NN,2,1,0,WORK)
      CALL ZERO(DATA)
      CALL FOURT(DATA,NN,2,-1,1,WORK)
C
      DO 80 I=1,128
       RMAT(I,1)=REAL(DATA(I,1))
       RMAT(I,2)=REAL(DATA(I,2))
80      CONTINUE
      CALL SCALE(RMAT,IMAT)
      DO 90 I=2,128
       ILND(1,1)=IMAT(I-1,1)
       ILND(1,2)=IMAT(I-1,2)
       ILND(2,1)=IMAT(I,1)
       ILND(2,2)=IMAT(I,2)
       CALL DRAW(ILND,IFLAG)
       IFLAG=0
```

```
90      CONTINUE
        RETURN
        END
```

```
C
      SUBROUTINE FFTBRD(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
      DIMENSION RMAT(128,2),IMAT(128,2),NN(2),WORK(50),IBRD(5,5)
      COMPLEX DATA(128,2)
      DATA NN/128,2/
C
      L=J
      IPTS=128
      ILEG=128/6
      RNTVL=.9/(IPTS-1)
C
      CALL OFST(L,RAMTAB,RAMTBL)
C
      X(1)=RAMTAB(J,1)
      Y(1)=RAMTAB(J,2)
C
      DO 10 I=2,ILEG
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)+RNTVL
10    CONTINUE
C
      K=ILEG+1
      KK=ILEG+(128/6)-1
C
      DO 20 I=K,KK
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
20    CONTINUE
C
      K=KK+1
      KK=KK+(128/6)-1
C
      DO 30 I=K,KK
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)-RNTVL
30    CONTINUE
C
      K=KK+1
      KK=KK+(128/6)-1
      X(K)=X(1)
      Y(K)=Y(K)+(42*RNTVL)
C
      K=K+1
      DO 40 I=K,KK
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)-RNTVL
40    CONTINUE
C
      K=KK+1
      KK=KK+(128/6)-1
C
      DO 50 I=K,KK
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
50    CONTINUE
```

```
C
      K=KK+1
      DO 60 I=K,IPT3
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)+RNTVL
 60   CONTINUE
C
      DO 70 I=1,128
       DATA(I,1)=CMPLX(X(I),0.0)
       DATA(I,2)=CMPLX(Y(I),0.0)
 70   CONTINUE
C
      CALL FOURT(DATA,NN,2,1,0,WORK)
      CALL ZERO(DATA)
      CALL FOURT(DATA,NN,2,-1,1,WORK)
C
      DO 80 I=1,128
       RMAT(I,1)=REAL(DATA(I,1))
       RMAT(I,2)=REAL(DATA(I,2))
 80   CONTINUE
C
      CALL SCALE(RMAT,IMAT)
C
      DO 90 I=2,128
       IBRD(1,1)=IMAT(I-1,1)
       IBRD(1,2)=IMAT(I-1,2)
       IBRD(2,1)=IMAT(I,1)
       IBRD(2,2)=IMAT(I,2)
       CALL DRAW(IBRD,IFLAG)
       IFLAG=0
 90   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FFTBLD(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
      DIMENSION RMAT(128,2),IMAT(128,2),NN(2),WORK(50),IBLD(5,5)
      COMPLEX DATA(128,2)
      DATA NN/128,2/
C
      L=J
      IPTS=128
      ILEG=128/7
      RNTVL=.9/(IPTS-1)
C
      CALL UFST(L,RAMTAB,RAMTBL)
C
      X(1)=RAMTBL(J,1)
      Y(1)=RAMTBL(J,2)
C
      DO 10 I=2,ILEG
       X(I)=X(I-1)
       Y(I)=Y(I-1)+RNTVL
10    CONTINUE
C
      K=ILEG+1
      KK=ILEG+(128/7)
C
      DO 20 I=K,KK
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
20    CONTINUE
C
      VALX=X(KK)
      VALY=Y(KK)
      K=KK+1
      KK=KK+(128/7)
C
      X(K)=VALX-(8*RNTVL)
      Y(K)=VALY
      K=K+1
C
      DO 30 I=K,KK
       X(I)=X(I-1)
       Y(I)=Y(I-1)-RNTVL
30    CONTINUE
C
      K=KK+1
      KK=KK+(128/7)
C
      X(K)=VALX
      Y(K)=VALY
C
      K=K+1
C
      DO 40 I=K,KK
       X(I)=X(I-1)+(0.5*RNTVL)
       Y(I)=Y(I-1)
40    CONTINUE
C
```

```
      K=KK+1
      KK=KK+(128/7)
C
      DO 50 I=K,KK
       X(I)=X(I-1)+(0.5*RNTVL)
       Y(I)=Y(I-1)
50    CONTINUE
C
      K=KK+1
      KK=KK+(128/7)
      DO 60 I=K,KK
      X(I)=X(I-1)
      Y(I)=Y(I-1)-(0.5*RNTVL)
60    CONTINUE
C
      K=KK+1
C
      DO 70 I=K,IPTS
       X(I)=X(I-1)-RNTVL
       Y(I)=Y(I-1)
70    CONTINUE
C
      DO 80 I=1,128
       DATA(I,1)=CMPLX(X(I),0.0)
       DATA(I,2)=CMPLX(Y(I),0.0)
80    CONTINUE
C
      CALL FOURT(DATA,NN,2,1,0,WORK)
      CALL ZERO(DATA)
      CALL FOURT(DATA,NN,2,-1,1,WORK)
C
      DO 90 I=1,128
       RMAT(I,1)=REAL(DATA(I,1))
       RMAT(I,2)=REAL(DATA(I,2))
90    CONTINUE
      CALL SCALE(RMAT,IMAT)
C
      DO 100 I=2,128
       IBLD(1,1)=IMAT(I-1,1)
       IBLD(1,2)=IMAT(I-1,2)
       IBLD(2,1)=IMAT(I,1)
       IBLD(2,2)=IMAT(I,2)
       CALL DRAW(IBLD,IFLAG)
       IFLAG=0
100   CONTINUE
      RETURN
      END
```

AD-A103 378 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
AN EVALUATION OF DETAIL IN DYNAMIC VISUAL DISPLAYS.(U)
DEC 80 M A SMART

UNCLASSIFIED AFIT/GCS/EE/80-14 NL

2 of 2

END
DATE
FILMED
0 81
DTIC

```
      SUBROUTINE FFTLLE(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
      DIMENSION RMAT(128,2),IMAT(128,2),ITLE(5,5),NN(2),WORK(50)
      COMPLEX DATA(128,2)
      DATA NN/128,2/
      L=J
      IPTS=128
      ILEG=128/3
      RNTVL=.9/(IPTS-1)
      CALL OFST(L,RAMTAB,RAMTBL)
C
      X(1)=RAMTBL(J,1)
      Y(1)=RAMTBL(J,2)
C
      DO 10 I=2,ILEG
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
10    CONTINUE
C
      K=ILEG+1
      KK=IPTS-ILEG-1
C
      X(K)=X(ILEG)+(0.5*RNTVL)
      Y(K)=Y(1)
C
      K=K+1
C
      DO 20 I=K,KK
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
20    CONTINUE
C
      K=KK+1
      X(K)=X(KK)+(0.5*RNTVL)
      Y(K)=Y(1)
      K=K+1
      DO 30 I=K,IPTS
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
30    CONTINUE
C
      DO 40 I=1,128
       DATA(I,1)=CMPLX(X(I),0.0)
       DATA(I,2)=CMPLX(Y(I),0.0)
40    CONTINUE
C
      CALL FOURT(DATA,NN,2,1,0,WORK)
      CALL ZERO(DATA)
      CALL FOURT(DATA,NN,2,-1,1,WORK)
C
      DO 50 I=1,128
       RMAT(I,1)=REAL(DATA(I,1))
       RMAT(I,2)=REAL(DATA(I,2))
50    CONTINUE
C
      CALL SCALE(RMAT,IMAT)
```

```
C
      DO 60 I=2,128
       IELE(1,1)=IMAT(I-1,1)
       IELE(1,2)=IMAT(I-1,2)
       IELE(2,1)=IMAT(I,1)
       IELE(2,2)=IMAT(I,2)
       CALL DRAW(IMAT,IFLAG)
       IFLAG=0
   60 CONTINUE
      RETURN
      END
```

```
C
C
      SUBROUTINE FFTPTR(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),RAMTBL(21,3),IPTR(5,5),X(128),Y(128)
      DIMENSION RMAT(128,2),IMAT(128,2),NN(2),WORK(50)
      COMPLEX DATA(128,2)
      DATA NN/128,2/
C
      L=J
      IPTS=128
      ILEG=128/3
      RNTVL=.9/(IPTS-1)
C
      CALL OFST(L,RAMTAB,RAMTBL)
      X(1)=RAMTBL(J,1)
      Y(1)=RAMTBL(J,2)
      DO 10 I=2,ILEG
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
10    CONTINUE
C
      J=ILEG
      K=ILEG+1
      KK=IPTS-ILEG-1
C
      DO 20 I=K,KK
       X(I)=X(I-1)-(0.5*RNTVL)
       Y(I)=Y(I-1)+(0.5*RNTVL)
20    CONTINUE
C
      K=KK+1
      Y(K)=Y(1)
      X(K)=X(J)
C
      K=K+1
      DO 30 I=K,IPTS
       X(I)=X(I-1)-(0.5*RNTVL)
       Y(I)=Y(I-1)+(0.5*RNTVL)
30    CONTINUE
C
      DO 40 I=1,128
       DATA(I,1)=CMPLX(X(I),0.0)
       DATA(I,2)=CMPLX(Y(I),0.0)
40    CONTINUE
C
      CALL FOURT(DATA,NN,2,1,0,WORK)
      CALL ZERO(DATA)
      CALL FOURT(DATA,NN,2,-1,1,WORK)
C
      DO 50 I=1,128
       RMAT(I,1)=REAL(DATA(I,1))
       RMAT(I,2)=REAL(DATA(I,2))
50    CONTINUE
C
      CALL SCALE(RMAT,IMAT)
      DO 60 I=2,128
```

```
      IPTR(1,1)=IMAT(I-1,1)
      IPTR(1,2)=IMAT(I-1,2)
      IPTR(2,1)=IMAT(I,1)
      IPTR(2,2)=IMAT(I,2)
      CALL DRAW(IPTR,IFLAG)
      IFLAG=0
60    CONTINUE
      RETURN
      END
```

```
C
C
      SUBROUTINE FFTARG(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,3),RAMTBL(21,3),X(128),Y(128)
      DIMENSION RMAT(128,2),IMAT(128,2),NN(2),WORK(50),ITARG(5,5)
C
      COMPLEX DATA(128,2)
      DATA NN/128,2/
C
      L=J
      IPTS=128
      ILEG=128/4
      RNTVL=.9/(IPTS-1)
C
      CALL OFST(L,RAMTAB,RAMTBL)
      X(1)=RAMTBL(J,1)+RNTVL
      Y(1)=RAMTBL(J,2)
C
      DO 10 I=2,ILEG
       X(I)=X(I-1)+RNTVL
       Y(I)=Y(I-1)
10    CONTINUE
C
      K=KK+1
      KK=IPTS-(ILEG*2)-1
      X(K)=X(1)-(2*RNTVL)
      Y(K)=Y(K-1)
      K=K+1
C
      DO 20 I=K,KK
       X(I)=X(I-1)-RNTVL
       Y(I)=Y(I-1)
20    CONTINUE
C
      K=KK+1
      KK=IPTS-ILEG-1
      Y(K)=Y(1)+RNTVL
      X(K)=X(1)
      K=K+1
C
      DO 30 I=K,KK
       X(I)=X(I-1)
       Y(I)=Y(I-1)+RNTVL
30    CONTINUE
C
      K=KK+1
      X(K)=X(1)
      Y(K)=Y(1)-RNTVL
      K=K+1
      DO 60 I=K,IPTS
       X(I)=X(I-1)
       Y(I)=Y(I-1)-RNTVL
60    CONTINUE
```

```
C
      DO 70 I=1,128
       DATA(I,1)=CMPLX(X(I),0.0)
       DATA(I,2)=CMPLX(Y(I),0.0)
   70 CONTINUE
C
C
      CALL FOURT(DATA,NN,2,1,0,WORK)
      CALL ZERO(DATA)
      CALL FOURT(DATA,NN,2,-1,1,WORK)
      DO 80 I=1,128
       RMAT(I,1)=REAL(DATA(I,1))
       RMAT(I,2)=REAL(DATA(I,2))
   80 CONTINUE
      CALL SCALE(RMAT,IMAT)
C
C
      DO 90 I=2,128
       ITARG(1,1)=IMAT(I-1,1)
       ITARG(1,2)=IMAT(I-1,2)
       ITARG(2,1)=IMAT(I,1)
       ITARG(2,2)=IMAT(I,2)
       CALL DRAW(ITARG,IFLAG)
       IFLAG=0
   90 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FFTHRD(J,RAMTAB,IFLAG)
      DIMENSION RAMTAB(21,5),NN(2),WORK(50),RMAT(128,2),X(128)
      DIMENSION Y(128),IBRID(5,5),IMAT(128,2),RAMTBL(21,3)
      COMPLEX DATA(128,2)
      DATA NN/128,2/
       IPTS=128
       ILEG=IPTS/3
       RNTVL=(.9/(IPTS-1))
      L=J
      CALL OFST(L,RAMTAB,RAMTBL)
       X(1)=RAMTBL(J,1)
       Y(1)=RAMTBL(J,2)
         DO 10 I=2,ILEG
          X(I)=X(I-1)+RNTVL
          Y(I)=Y(I-1)
   10    CONTINUE
        KK=IPTS-ILEG-1
        K=ILEG+1
          DO 20 I=K,KK
           X(I)=X(I-1)+RNTVL
           Y(I)=Y(I-1)
   20     CONTINUE
         K=KK+1
          DO 30 I=K,IPTS
           X(I)=X(I-1)+RNTVL
        Y(I)=Y(I-1)
   30  CONTINUE
        DO 50 J=1,128
          DATA(J,1)=CMPLX(X(J),0.0)
          DATA(J,2)=CMPLX(Y(J),0.0)
   50    CONTINUE
        CALL FOURT(DATA,NN,2,1,0,WORK)
        CALL ZERO(DATA)
        CALL FOURT(DATA,NN,2,-1,1,WORK)
          DO 60 I=1,128
           RMAT(I,1)=REAL(DATA(I,1))
           RMAT(I,2)=REAL(DATA(I,2))
   60   CONTINUE
         CALL SCALE(RMAT,IMAT)
          DO 70 I=2,128
             IBRID(1,1)=IMAT(I-1,1)
             IBRID(1,2)=IMAT(I-1,2)
             IBRID(2,1)=IMAT(I,1)
             IBRID(2,2)=IMAT(I,2)
             CALL DRAW(IBRID,IFLAG)
            IFLAG=0
   70      CONTINUE
         RETURN
         END
```

```
        .TITLE  PRSPCT
        .GLOBL  PRSPCT
        .GLOBL  INIT            ;INITIALIZATION FOR INTERACTIVE MAP
        .GLOBL  INWIND          ;INITIALIZATION OF WINDOW
        .GLOBL  NAV             ;NAVIGATION ROUTINE
        .GLOBL  OPVRIN          ;OPEN VERTICAL DATA FILE FOR INPUT
        .GLOBL  OPHRIN          ;OPEN HORIZONTAL DATA FILE FOR INPUT
        .GLOBL  CLVRIN          ;CLOSE VERTICAL FILE FROM INPUT
        .GLOBL  CLHRIN          ;CLOSE HORIZONTAL FILE FROM INPUT
        .GLOBL  WINDOW          ;UPDATE THE TOROIDAL MEMORY WINDOW
        .GLOBL  FRAME           ;GENERATE THE PERSPECTIVE VIEW
        .GLOBL  BUFFER          ;RAMTEK BUFFER NUMBER PARAMTER
        .MCALL  EXIT$S,SPWN$S
SYMBOL: .ASCII  /SYMBOL/
GC:     .WORD   2
MC:     .WORD   3
EFN:    .WORD   4
PRSPCT: JSR     R5,INTER
        JSR     R5,INIT         ;INITIALIZE PARAMETERS FOR INTERACTIVE MAP
        JSR     R5,OPVRIN       ;OPEN THE VERTICAL FILE FOR INPUT
        JSR     R5,OPHRIN       ;OPEN THE HORIZONTAL FILE FOR INPUT
        JSR     R5,INWIND       ;INITIALIZE THE TOROIDAL MEMORY WINDOW
        MOV     #0.,INIRAM+10   ;THESE ARE DEFAULT VALUES - THEY CAN
        MOV     #0,INIRAM+12    ;BE REMOVED FROM PRCOM
        MOV     #639.,INIRAM+14
        MOV     #479.,INIRAM+16
        MOV     #INIRAM,BUFFER  ;SET PARAMETER TO INITIALIZE THE RAMTEK
        JSR     R5,RAMTEK               ;INITIALIZE THE RAMTEK
        MOV     #PLANE,BUFFER   ;DRAW THE AIRPLANE SYMBOL
        JSR     R5,RAMTEK
        MOV     #0,BUF3
        MOV     #17,BUF3+100
        MOV     #360,BUF3+200
        MOV     #7760,BUF3+776
        MOV     #COLOR,BUFFER
        JSR     R5,RAMTEK
        MOV     #0,AGL+16       ;RESTORE THE LOWER LEFT HAND CORNER POINTER
LOOP:   JSR     R5,JOYSTK
        JSR     R5,NAV          ;NAVIGATE TO NEW POSITION
        JSR     R5,WINDOW       ;UPDATE THE TOROIDAL MEMORY WINDOW
        JSR     R5,FRAME        ;CREATE THE PERSPECTIVE VIEW
        SPWN$S  #SYMBOL,#GC,#MC,#EFN
        TST     $DSW
        BLT     ERR
        BR      LOOP            ;LOOP FOREVER
ERR:    JSR     R5,CLVRIN       ;CLOSE THE VERTICAL FILE
        JSR     R5,CLHRIN       ;CLOSE THE HORIZONTAL FILE
        EXIT$S                  ;BYE-BYE!
        .END    PRSPCT
```

TEST RESULTS

The data that follows is the result of input to the thesis software. The input is data that is simulated to be like information that will be received from PRSPCT. The output statement, DIRECTIVE STATUS 1 and IOSTAT 1 indicates that the Ramtek driver sucessfully output information to the video display.

The first nine sets of data show that all symbols can be drawn. The second set of data shows that the program does not fail at the boundaries.

The corner coordinates are the coordinates of A, B, C, and D of the window. XMIN, XMAX, YMIN, and YMAX show the minimum and maximum boundary values. The other data is self-explanatory.

```
AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  10.50 INITIAL LONGITUDE:  15.50

CORNER COORDINATES
   15.50000   10.50000   15.21940   10.78060   15.50000   11.

   15.78060   10.78060

XMIN,XMAX,YMIN,YMAX
   15.21940   15.78060   10.50000   11.06120

 THE RAMTEK TABLE
   15.5000    10.7517    20.0000
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
HIGHWAY HAS BEEN  DRAWN
>

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  30.50 INITIAL LONGITUDE:  35.50

CORNER COORDINATES
   35.50000   30.50000   35.21940   30.78060   35.50000   31.

   35.78060   30.78060

XMIN,XMAX,YMIN,YMAX
   35.21940   35.78060   30.50000   31.06120

 THE RAMTEK TABLE
   35.5000    30.7517    40.0000
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
BRIDGE A HAS BEEN DRAWN

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  20.50 INITIAL LONGITUDE:  25.50

CORNER COORDINATES
   25.50000   20.50000   25.21940   20.78060   25.50000   21.

   25.78060   20.78060

XMIN,XMAX,YMIN,YMAX
   25.21940   25.78060   20.50000   21.06120

 THE RAMTEK TABLE
   25.5000    20.7517    30.0000
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
DIRECTIVE STATUS=       1
IOSTAT =     1
ELECTRICAL LINES HAVE BEEN DRAWN
```

```
80,85,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  80.50 INITIAL LONGITUDE:  85.50

CORNER COORDINATES
   85.50000   80.50000   85.21940   80.78060   85.50000   81.

   85.78060   80.78060

XMIN,XMAX,YMIN,YMAX
   85.21940   85.78060   80.50000   81.06120

 THE RAMTEK TABLE
   85.5030    80.7517    80.0000
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
>


40,45,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  40.50 INITIAL LONGITUDE:  45.50

CORNER COORDINATES
   45.50000   40.50000   45.21940   40.78060   45.50000   41.

   45.78060   40.78060

XMIN,XMAX,YMIN,YMAX
   45.21940   45.78060   40.50000   41.06120

 THE RAMTEK TABLE
   45.5030    40.7517    50.0000
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
DIRECTIVE STATUS=       1
IOSTAT -     1
BRIDGE 8 HAS BEEN DRAWN
>
```

```
30,35,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  30.50 INITIAL LONGITUDE:  35.50

CORNER COORDINATES
   35.50000   30.50000   35.21940   30.78000   35.50000   31.

   35.78000   30.78060

XMIN,XMAX,YMIN,YMAX
   35.21940   35.78060   30.50000   31.06120

 THE RAMTEK TABLE
   35.5000    30.7517    40.0000
DIRECTIVE STATUS=      1
IOSTAT -     1
DIRECTIVE STATUS=      1
IOSTAT -     1
DIRECTIVE STATUS=      1
IOSTAT -     1
BRIDGE A HAS BEEN DRAWN
>
60,65,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  60.50 INITIAL LONGITUDE:  65.50

CORNER COORDINATES
   65.50000   60.50000   65.21940   60.78060   65.50000   61.

   65.78060   60.78050

XMIN,XMAX,YMIN,YMAX
   65.21940   65.78060   60.50000   61.06120

 THE RAMTEK TABLE
   65.5000    60.7617    70.0000
DIRECTIVE STATUS=      1
IOSTAT -     1
DIRECTIVE STATUS=      1
IOSTAT -     1
DIRECTIVE STATUS=      1
IOSTAT -     1
THE LANDING STRIP HAS BEEN DRAWN
>

50,55,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  50.50 INITIAL LONGITUDE:  55.50

CORNER COORDINATES
   55.50000   50.50000   55.21940   50.78060   55.50050   51.

   55.78060   50.78060

XMIN,XMAX,YMIN,YMAX
   55.21940   55.78060   50.50000   51.06120

 THE RAMTEK TABLE
   55.5000    50.7517    60.0000
DIRECTIVE STATUS=      1
IOSTAT -     1
DIRECTIVE STATUS=      1
IOSTAT -     1
DIRECTIVE STATUS=      1
IOSTAT -     1
THE POINTER HAS BEEN DRAWN
>
```

```
70,75,30.,30.,0.,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
  0 DEGREES.   1. KNOTS

INITIAL LATITUDE:  70.50 INITIAL LONGITUDE:  75.50

CORNER COORDINATES
   75.50000   70.50000   75.21940   70.70060   75.50000   71.

   75.78060   70.78060

XMIN,XMAX,YMIN,YMAX
   75.21940   75.78060   70.50000   71.06120

 THE RAMTEK TABLE
   75.5000    70.7517    80.0000
DIRECTIVE STATUS=      1
IOSTAT =     1
DIRECTIVE STATUS=      1
IOSTAT =     1
DIRECTIVE STATUS=      1
IOSTAT =     1
DIRECTIVE STATUS=      1
IOSTAT =     1
THE TARGET SYMBOL HAS BEEN DRAWN
>
```

8?,35,0.,0.,1.104,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
45 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES
35.00000 85.00000 35.12430 85.37685 35.50116 85.

35.37686 84.87569

XMIN,XMAX,YMIN,YMAX
35.00000 35.50116 84.87569 85.37685

85,35,0.,0.,2.209,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
90 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES
35.00000 85.00000 35.39254 85.05820 35.45074 84.

35.05820 84.60745

XMIN,XMAX,YMIN,YMAX
35.00000 35.45074 84.60745 85.05820

85,35,0.,0.,3.313,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
135 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES
35.00000 85.00000 35.22852 84.67565 34.90428 84.

34.67565 84.77138

XMIN,XMAX,YMIN,YMAX
34.67565 35.22852 84.44703 85.00000

85,35,0.,0.,4.414,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
180 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES
35.00000 85.00000 34.81429 84.64931 34.46360 84.

34.64931 85.18572

XMIN,XMAX,YMIN,YMAX
34.46360 35.00000 84.64931 85.18572

85,35,0.,0.,5.515,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
225 DEGREES. 1. KNOTS

INITIAL LATITUDE: 85.00 INITIAL LONGITUDE: 35.00

CORNER COORDINATES
35.00000 85.00000 34.60323 85.00584 34.61006 85.

35.00683 85.39677

XMIN,XMAX,YMIN,YMAX
34.60323 35.00683 85.00029 85.40359

```
85,5\5\35,0.,0.,6.616,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
270 DEGREES.    1. KNOTS

INITIAL LATITUDE:  85.00 INITIAL LONGITUDE:  35.00

CORNER COORDINATES
   35.00000   85.00000   34.82647   85.35688   35.10335   85.

   35.35687   85.17353

XMIN,XMAX,YMIN,YMAX
   34.82647   35.35687   85.00000   85.53041



85,35,0.,0.,7.720,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
315 DEGREES.    1. KNOTS

INITIAL LATITUDE:  85.00 INITIAL LONGITUDE:  35.00

CORNER COORDINATES
   35.00000   85.00000   35.24060   85.31557   35.55617   85.

   35.31557   84.75941

XMIN,XMAX,YMIN,YMAX
   35.00000   35.55617   84.75941   85.31557



85,35,0.,0.,8.824,10.,10.,64,64,0.,1.

AIRCRAFT HEADING AND SPEED
360 DEGREES.    1. KNOTS

INITIAL LATITUDE:  85.00 INITIAL LONGITUDE:  35.00

CORNER COORDINATES
   35.00000   85.00000   35.39009   84.92716   35.31724   84.

   34.92716   84.60932

XMIN,XMAX,YMIN,YMAX
   34.92716   35.39009   84.53706   85.00000



75,70,0.,0.,1.104,10.,10.,64,64,2.209,1.

AIRCRAFT HEADING AND SPEED
135 DEGREES.    1. KNOTS

INITIAL LATITUDE:  75.00 INITIAL LONGITUDE:  70.00

CORNER COORDINATES
   70.00000   75.00000   70.22863   74.67565   69.90427   74.

   69.67566   74.77133

XMIN,XMAX,YMIN,YMAX
   69.67566   70.22863   74.44703   75.00000
```

VITA

Mary A. Smart was born 24 February, 1957 in Newark, Ohio. She graduated from Newark Catholic High School in Newark, Ohio in June, 1975. She attended Ohio Dominican College, Columbus, Ohio where she graduated in May, 1979 with a Bachelor of Arts degree in Mathematics and Business Administration. While attending Ohio Dominican College, where participated in the Reserve Officers Training Program at Capital University, Columbus, Ohio. She was commissioned as a second Lieutenant in the United States Air Force in May, 1979. In June, 1979 she was assigned to the Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Permanent address:38 MadisonAvenue
Newark, Ohio 43055

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFIT/GCS/EE/80D-14	2. GOVT ACCESSION NO. AD-A103 378 / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Evaluation of Detail in Dynamic Visual Displays	5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mary A. Smart 2Lt USAF	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-ENA) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Information, Presentation, and Controls Group, Air Force Avionics Lab (AFWAL-AAAT-1) Wright-Patterson, AFB, OHIO 45433	12. REPORT DATE 1 December 1980 13. NUMBER OF PAGES 113
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

21 AUG 1981

18. SUPPLEMENTARY NOTES

Frederic C. Lynch

Frederic Lynch Major USAF

Director of Public Affairs ~~20 JUL 1981~~

Distribution unlimited
IAW-AFR 190-17

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Airborne Electronic Terrain Mapping System (AETMS)
symbology
basic symbol set
visual display Fast Fourier Transform (FFT)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Airborne Electronic Terrain Mapping System (AETMS) is a mapping system designed to be used in aircraft. The report explains what software is necessary to provide an overlay of symbols onto the terrain map. The report also suggests a symbol and color set for the AETMS as well as a format to test the symbol set. Fast Fourier Transforms (FFT's) are a sub-set of the test symbol set.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)